

Kodak

Info Input Solution

Info Input Solution

Administrator's Guide

Version 7.0



Table of Contents

Legal Notice	1
1. Welcome	2
1.1. Architectural Overview	2
1.2. Requirements for Info Input Solution Application Server	3
1.3. System Administration Overview	5
2. Getting started with Info Input Solution	7
2.1. Clients	7
2.2. Info Input Solution Mobile app	16
3. Jobs Setup	18
3.1. Jobs	21
3.2. Shared Objects	190
3.3. Scripting	233
4. Scan Profiles Management	239
4.1. Defining Scan Profiles	239
4.2. Global Scan Profiles	242
4.3. Selecting Scan Profiles	245
4.4. Scan Profiles and New Batches	246
5. Import Service Administration	249
5.1. Start the Import Service	249
5.2. Launch the Import Service Administration window	249
5.3. File System Import	251
5.4. XML File Import	254
5.5. Email Import	255
5.6. Box Skill Import	258
6. Export Destinations Management	261
6.1. Export Destination Templates	262
6.2. Editing Export Destinations	263
6.3. Images Export	265
6.4. Kofax Export	269
6.5. File / folder Move	270
6.6. Disk Export	271
6.7. Database Export	271

6.8. ECM Export	273
6.9. Scripting Export	275
6.10. CSV Export	278
7. Batch and Workflow Management	280
7.1. Batch Manager	280
7.2. Performing Actions on Batches	281
7.3. Customizing the Batch Manager	285
7.4. Queues	291
7.5. Private Batches	293
8. User Management	296
8.1. User and Group Administration	296
8.2. Authentication and Authorization	315
8.3. Licensing Administration	318
8.4. Active Users	318
8.5. Offline Scanning	319
8.6. Announcements	321
9. Task Filters	327
9.1. Configuring and customizing Task Filters	327
10. Custom Fields	332
10.1. Custom Fields Instructions	332
11. Home Tab	336
11.1. Instructions	337
11.2. Custom Home Tab Examples	340
12. Server Administration	343
12.1. Administration Utility	343
12.2. Server Configuration Parameters	344
12.3. Debug Client	353
12.4. Server Configuration properties	355
13. Intelligent REST Services	367
Appendix A: Regex Search	369
A.1. Introduction	369
A.2. Regular Expression Syntax	369
A.3. Limitations	369
A.4. Technical Details	369

Legal Notice

Copyright © 2008-2024 Kodak Alaris Inc.. All rights reserved. Info Input Solution is a trademark or registered trademark of Kodak Alaris Inc.. The list of trademarks is not exhaustive of other trademarks, registered trademarks, product names, company names, brands and service names mentioned herein are property of Open Text or other respective owners.

Disclaimer

No Warranties and Limitation of Liability.

Every effort has been made to ensure the accuracy of the topics presented in this publication. However, Kodak Alaris Inc. and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.



This software is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

1. Welcome

This Administrator's Guide contains the information needed to set up scanning, indexing and export for Jobs in *Info Input Solution*. It is intended to be used by the administrators of *Info Input Solution*. This guide assumes the reader has basic working knowledge of generic Document Imaging software terms and Enterprise Content Management procedures.

1.1. Architectural Overview

The *Info Input Solution* system is unique in that, all communication between the client and the server is over a single HTTP or HTTPS connection, utilizing Web Services. This architecture provides the best and easiest way to secure the application and provide high availability using standard techniques and devices.

The server component of the system is designed as a standard J2EE application.

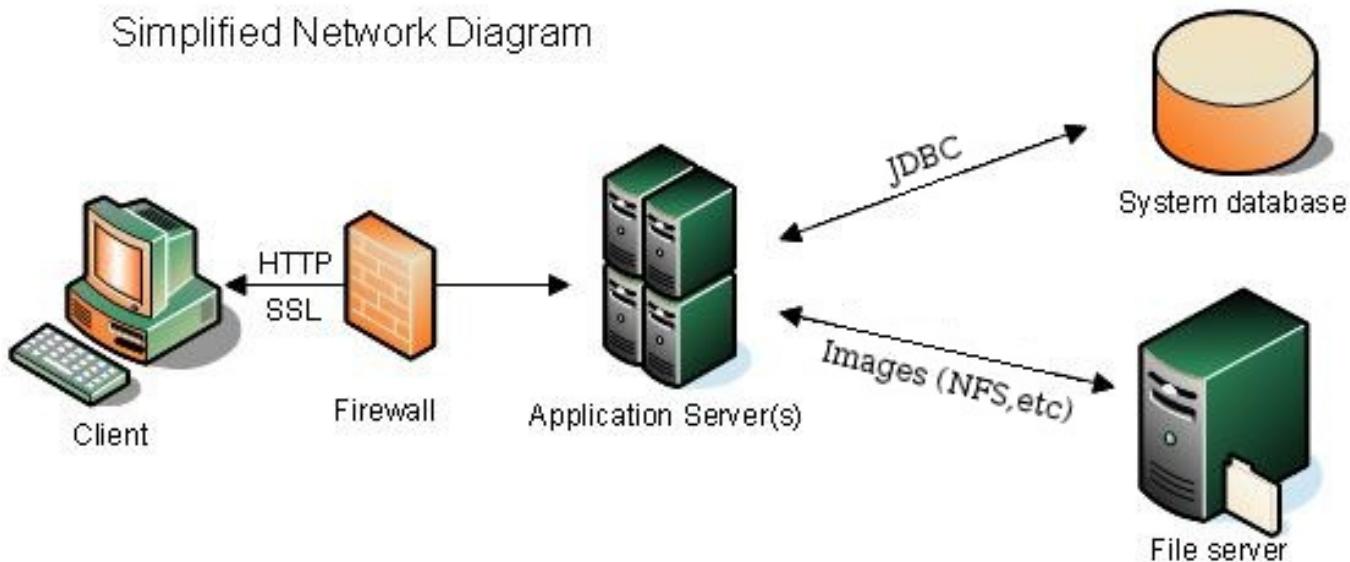


Figure 1. Architectural Overview

This architecture provides the following benefits:

- There are no special network connectivity requirements other than standard web access. If a client can access the web, it can successfully run the *Thick Client* or the *HTML Client*.
- All communication can be secured using standard SSL encryption.
- No configuration in corporate firewalls is required. No custom ports need to be opened.
- The application can be integrated immediately in an existing corporate application server, utilizing all security schemes and standards already in place.

Info Input Solution supports MS SQL Server, Oracle Database, or PostgreSQL for the system database. The images are stored separately on the file system (local or network).

1.2. Requirements for Info Input Solution Application Server

1.2.1. System requirements for the Info Input Solution Application Server

System Sizing

- Small Volume: 4 concurrent users / 200K pages per year
- Medium Volume 80 users / 10M pages per year
- High Volume: 250 users / 50M pages per year

CPU

- Minimum requirement is 6 cores at 2GHz.
- For High Volume systems, the recommended configuration is 32 cores @2.7+ GHz. Contact the Support team for the exact requirements according to the business case.
- When the Info Input Solution HTML Client is used by end-users, additional CPU is required for the HTML supporting service/process. The hardware specs depend on the use case and estimated load.
- When the Info Input Solution Classification and Extraction server-side engines are used in the Job Workflow(s), additional CPU is required for the Info Input Solution Core service/process. The hardware specs depend on the use case and estimated load.
- For virtual environments, features like Intel VT-x / VT-d and AMD V / Vi must be supported by the processors, activated on the machine, enabled in the Virtualization product and activated for the virtual machine.

Memory

- Minimum memory size is 6 GB.
- Recommended memory size for High Volume systems is 32 GB.

Operating System

- Windows Server (on premises or Cloud) 2012 / 2016 / 2019 / 2022.
- Red Hat Enterprise Linux 7 / 8.
- SUSE Linux Enterprise Server 12 / 15.

Disk Storage

- Single server installation: The recommended configuration is a dedicated disk array (RAID 5 or 10) either physically installed on the server, or exported over a dedicated high-speed Storage Area Network (SAN). The minimum configuration is a single disk, again either physically installed in the server, or exported over a SAN.
- Cluster installation: The recommended configuration is a dedicated Network Attached Storage

(NAS) unit, using a file system native to the OS of the application servers, connected to the server over a dedicated high-speed network. The minimum configuration is a NAS that can be used by all nodes in the server cluster.

- SMB v3 compatible Windows Server file storage systems are supported.

Network

- Single server: The recommended configuration is separate networks for HTTP, Database and storage traffic, if external storage is used. All networks should be at least 1 Gb and all nodes should be attached to the networks through switches. The minimum configuration for Small Volume systems is a single 1 Gb network.
- Server cluster: In addition to the single-server recommended configuration, the recommended configuration for a cluster is to provide additional networks for cluster heartbeat and reverse proxy to cluster node HTTP traffic. The minimum configuration for Small Volume systems is a single 1 Gb network.

1.2.2. Software requirements for the Info Input Solution Application Server

Apache Tomcat: version 9.0.84 64-bit (included in the installation package) or any 9.0.x Tomcat version.

Database

- SQL Server 2014 / 2016 / 2017 / 2019 / 2022.
- Oracle 11.2 / 12.1 / 18c / 19c.
- PostgreSQL 13 or later.
- Azure SQL Server.

Bundled Database in installer

If during installation, you choose to install a new instance of SQL Server 2019 Express, the operating system must meet the minimum requirements:

- SQL Server 2019 installer will automatically install/enable Microsoft .NET Framework 4.6.
- Windows Server 2012 R2 require KB2919355 before installing .NET Framework 4.6.

For more information, see Hardware and Software Requirements for installing SQL Server 2017 at the Microsoft website.

1.2.3. Antivirus detection/interference

It is possible that an Antivirus program might detect Info Input Solution or any of its components as a threat. Which would in turn cause instability and block the seamless functioning of the application. In that case it is recommended to whitelist all the directories used by Info Input Solution as well as the opener on the respective workstations. This includes:

- The installation directory.

- \Users\<Current_user>\.scanclient
- \Users\<Current_user>\AppData\Local\ Info Input Solution
- The opener executable on the respective workstations where Info Input Solution will be used.

1.2.4. Requirements for Thick Client

Before launching the Thick Client, make sure the user's workstation meets the following pre-requisites.

- Operating Systems: All supported Windows OS versions (Windows 8.1, 10, 11, 2012, 2016, 2019 and 2022, Windows 10 or 11 at Windows 365 Azure Cloud PC). The Thick Client can also be accessed through the Microsoft Azure Virtual Desktop application.
- All contemporary web browsers are supported: Google Chrome, MS Edge, Mozilla Firefox, Mozilla Firefox ESR.
- With version 7.0, running the Application as an Applet inside a browser is no longer supported.

1.2.5. Requirements for HTML Client

Before launching the HTML Client application, make sure the user's workstation meets the following pre-requisites.

- Operating Systems: All supported Windows OS versions (Windows 8.1, 10, 11, Windows Server 2012, 2016, 2019 and 2022, Windows 10 or 11 at Windows 365 Azure Cloud PC). The HTML Client can also be accessed through the Microsoft Azure Virtual Desktop application. All supported Mac OS versions.
- All contemporary web browsers are supported: Google Chrome v40+, MS Edge, Mozilla Firefox 40+, Mozilla Firefox ESR 40+, Apple Safari (on Mac OS).

1.3. System Administration Overview

Before beginning the administrative tasks in *Info Input Solution*, an administrator needs to be designated. The *Info Input Solution* administrator will be responsible for the following system configurations/tasks:

Job Setup

Info Input Solution uses *Jobs* to define the process flow/settings for the system.

Scan Profiles Management

Info Input Solution uses *Scan Profiles* for the scanner configurations.

Export Destinations Management

The defined *Export Destinations* are used for exporting *Batch* output data to external systems.

Batch and Workflow Management

Info Input Solution uses *Batch Manager* to review/manage the work available in the system.

User Management

The *Info Input Solution* administrator designs the security model by setting up users and groups, and establishing their roles and security privileges.

Server Administration

The *Info Input Solution* administrator manages the database configuration and imported licenses, and configures various server parameters.

Announcements

Custom announcements that will appear in the user's screen after logging in can be managed by the *Info Input Solution* administrator.

Licensing

The *Info Input Solution* administrator can manage the mobile licenses

2. Getting started with Info Input Solution

Each user accesses *Info Input Solution* via a web browser, using a login ID and password. *Info Input Solution* can be configured to store user ID's and passwords internally in the central database, authenticate users via Windows Active Directory, or be integrated to a Single Sign-on system.

2.1. Clients

2.1.1. Thick Client

The Thick Client can be started as a stand-alone application. To launch the Thick Client as a stand-alone application follow the steps below:

1. Launch one of the following browsers (Google Chrome, MS Edge, MS Internet Explorer, Mozilla Firefox, Mozilla Firefox ESR) and navigate to *Info Input Solution*.

```
http://<server>:<port>/client/
```

For example

```
http://localhost:8080/client/
```

2. This will open the landing page of *Info Input Solution*. Click on Download *Info Input Solution* to open the Thick Client.

Kodak Info Input Solution

Launch Info Input Solution

Is Info Input Solution not installed? [Download Info Input Solution](#)

Looking for 64-bit? Launch the [64-bit](#) version

Figure 2. Landing Page

3. Then run the downloaded iis-opener executable to launch the *Info Input Solution* Thick Client. The filename of the iis-opener executable will have a form like the one below:

 iis-opener#Uiisopener%3Ahttps%3A%2F%2F192.168.2.181%3A443%2Fclient%2Fopener.jsp%3Fconfig%3Dopener.32.ini%26#.exe

Figure 3. Opener file

4. The next time the user will open the Thick Client URL a prompt will appear that will ask to automatically open the Thick Client.

The first *Thick Client* screen is the login dialog. Provide your authentication credentials to log in to the *Info Input Solution*. If *Windows Active Directory* is configured, check the *Windows Login Credentials* option instead to enable automatic logon.



Figure 4. Login dialog

Almost every administrator function described in this guide requires administrative privileges. The default administrator credentials for *Info Input Solution* are:

user: *admin*

password: <empty>

If this is the first time you login to *Info Input Solution*, make sure you setup a new non-empty password for the admin user, via the *Tools & Options* menu (≡) → *Change password...* item.

2.1.1.1. Thick Client configuration parameters

Everytime the *Info Input Solution* user attempts to launch the *Thick Client*, the *Info Input Solution* Opener mechanism communicates with the Apache Tomcat service and checks if there is a newer *Thick Client* version: in case there is a newer version, the *Info Input Solution* Opener will automatically update all *Thick Client* files with the latest version.

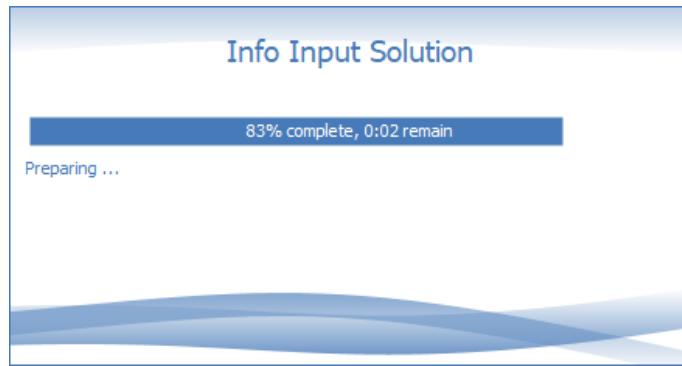


Figure 5. Loading dialog

On the server side, the files `opener.32.ini` and `opener.64.ini` (located under the `client/` folder) are the default *Thick Client* configuration files for 32bit and 64bit architectures. These files can be used to modify the startup behavior of the *Thick Client* when opened using the *opener* mechanism. Specifically the *initial-heap-size* and *max-heap-size* parameters can be used to modify the default memory settings of the *Thick Client*, the *opener_architecture* defines the *Thick Client* architecture and *opener_create_shortcut* defines if a *Thick Client* desktop shortcut will be created. According to the architecture defined in these files the appropriate 32bit or 64bit Opener web page will be loaded when the user will hit the *Thick Client* URL on the browser. Custom file with the above parameters can be created and provide it to the user as the *Thick Client* URL using the *config* URI parameter.

For example:

```
http://<server>:<port>/client/?config=customFilename.ini
```

There are also additional URI parameters that can be used by the *Info Input Solution* administrator. The *opener_architecture* URI parameter will define the default *Thick Client* architecture and it will load the appropriate Opener web page. The *proxy* parameter defines the forward proxy server and port configured on user's workstation. Do not use the *proxy* URI parameter, if a reverse proxy server is used in between the *Info Input Solution* Client workstation and the *Info Input Solution* server. *Info Input Solution* configuration details for a reverse proxy, are described in the *Installation Guide* at chapter 6.1.1.

Architecture example:

```
http://<server>:<port>/client/?opener_architecture=64
```

Proxy example:

http://<server>:<port>/client/?proxy=forwardProxyServer:8558

A long domain name and long proxy server name that create a URL string longer than 150 encoded characters it will be encoded using chinese characters and the users will see the Opener executable file-name like in the screenshot below. This is the expected behaviour. In case the encoded URL string along with the Windows path is over 255 characters the *Info Input Solution* users will not be able to launch the *Thick Client*. This is an MS Windows path length limit which prevents the executable to run:



Figure 6. Opener executable

Additional global Client parameters can be configured by the administrator, using the [Server Administration](#) menu.

2.1.1.2. Multi-lingual User Interface

The *Thick Client* is available in the following languages:

- English
- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- French
- German
- Japanese
- Polish
- Spanish
- Turkish

At first login, the *Thick Client* application will detect the local operating system language settings and will try to match the user preferred language and display locale settings. After successful login, a user

can change the *Thick Client* display language via the *Tools & Options Menu* → *Languages* menu item. The user preference will be stored locally, and future launches of the *Thick Client* will use the latest display language selection.

Note that, the *Info Input Solution* Installation Wizard and the [Administration Utility](#) (found in Start menu → *Info Input Solution* folder) are available only in English.

2.1.1.3. Thick Client Main Window

At the top of the main window is the main toolbar, with the following tools:

Tool	Name	Action
	<i>Batch menu</i>	Provide access to the <i>Batch menu</i> . This menu can also be accessed while working on an open batch, by right-clicking on the <i>Batch node</i> (root) in the <i>Batch Explorer Tree</i> .
	<i>New Batch</i>	Create a new batch. If a current open batch exists, the user is asked to close, suspend or delete it before proceeding.
	<i>Suspend Current Batch</i>	Temporarily suspend the current batch and leave it in the current queue.
	<i>Close Current Batch</i>	Close the current batch and upload it to the Core Service.
	<i>Scan</i>	Start scanning.
	<i>View</i>	View the current batch.
	<i>Index</i>	Index the current batch.
	<i>Exit</i>	Log-out and exit <i>Info Input Solution</i> .
	<i>Tools & Options</i>	View additional administrative tools and options to configure the user interface.

2.1.2. HTML Client

The HTML Client can be started by following the steps below,

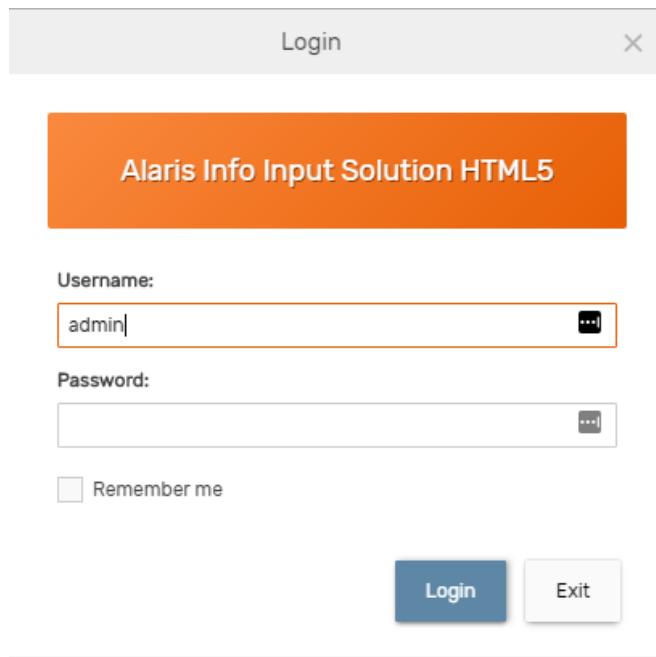
1. Launch one of the following browsers (Google Chrome, MS Edge, MS Internet Explorer, Mozilla Firefox, Mozilla Firefox ESR) and navigate to Info Input Solution HTML page.

http://<server>:<port>/client-html/

For example

http://localhost:8080/client-html/

2. The log in dialog below will appear, provide your authentication credentials to log in to the *HTML Client*.



2.1.2.1. HTML Client Main Window

At the top of the main window is the main toolbar, with the following tools:

Tool	Name	Action
	<i>Batch menu</i>	Provide access to the <i>Batch menu</i> . This menu can also be accessed while working on an open batch, by right-clicking on the <i>Batch node</i> (root) in the <i>Batch Explorer Tree</i> .
	<i>Scan</i>	Start scanning.
	<i>Index</i>	Index current batch
	<i>Exit</i>	Log-out and exit <i>Info Input Solution</i> .

Tool	Name	Action
	Pending Tasks	Shows if there are any pending tasks in the HTML support service.
	Tools & Options	View additional administrative tools and options.

2.1.2.2. Supported Features

The following features are supported in the *HTML Client*

- Scanning
- Indexing
- Manage Batches (Open, Close, Suspend, Delete)
- Batch Manager
- Managing of local Scan Profiles (create, delete, edit)
- Client-side Task Filtering
- Split Batch
- Node reviews

The following features are only supported from the *Thick Client*

- Windows integration for User Authentication: only supported by the *Thick Client*, due to HTML technology restrictions (not possible to retrieve OS level information for logged-in user session).
- Announcements
- Offline session
- Batch separation

2.1.2.3. HTML Client security features

HTML Service URL

The SVCURL configuration has been changed in 7.0 Instead of editing the `client-html/index.jsp` and changing the "srvBaseUrl": ... now is as follows:



if the customer svc-html URL domain and port is different from the cl-html then the following configuration steps must be taken:

1. Edit the `<installation directory>/client-html/index.jsp` file
2. Uncomment the `<!-- <ITConfiguration srvurl="http://127.0.0.1:8080"></ITConfiguration>`
--> line by removing the `<!--` at the start and the `-->` at the end of the line

3. Replace the `http://127.0.0.1:8080` with the actual HTML Service URL. i.e., `http://127.0.0.1:8080` → `http://domain.com:8080`
4. Save the file

Content Security Policy

Content Security Policy or CSP, is a built in security layer mechanism in web browsers which blocks the loading and executing of resources (scripts, images etc.) based on a given set of rules (the policy).

HTML Client supports the following (minimum) policy: `default-src https://domain.com:8080/service-html 'self'; script-src 'sha256-HASH' 'strict-dynamic' 'self'; font-src data: 'self'; style-src 'unsafe-inline' 'self'; img-src blob: data: 'self'`



the `https://domain.com:8080` is to be replaced with the actual URL of the HTML Service used in production. and sha256-HASH is to be replaced with the actual sha256 hash of the index.js file.

Enabling CSP

CSP can be enabled for the HTML Client by following these steps:

1. Stop the Core Service
2. Edit the `<installation directory>/config/client-html.response-header-filter` file
3. Locate the line containing the `<!--<header key="Content-Security-Policy"...`
4. Uncomment it by removing the `<!--` at the start and the `-->` at the end of the line
5. Replace the `URL` word inside the line with the actual URL of the HTML Service i.e., change `URL` → `http://domain.com:8080/service-html`
6. Start the Core Service



This CSP can be enriched with extra compatible custom rules. e.g. to add HTTP headers that allow CSP Reporting.

Details about the rules

`default-src`

The URL of the HTML Service is required to be listed because the HTML Client downloads various static files from the HTML Service e.g., configuration files, language and regional settings files etc.

`script-src`

The `'strict-dynamic'` value is required. This value allows one whitelisted javascript file to dynamically load and execute other javascript files without those having to be explicitly whitelisted. This

value is required by the HTML Client because it is not predictable which scripts will be loaded during a user session and also because the scripts can change in the midterm (e.g. updating jobs). It is also required to whitelist, using a hash, the javascript file index.js, which is loaded and executed first and loads every other javascript file. This is linked in the index.jsp using a <script> tag.

style-src

The value '**unsafe-inline**' is required. It is necessary because the HTML Client dynamically "domifies" HTML templates and dynamically mutates the .style property of DOM Elements for rendering and displaying various user interface elements.



The use of **unsafe-inline** for **style-src** with HTML Client actually poses no security issues because it sanitizes user input and does not allow them to inject custom styles.

img-src

The URL of the HTML Service is required to be listed because HTML Client downloads rubber-stamp image files for annotations from the HTML Service. The **data:** and **blob:** values are required. This rule controls which image sources the DOM elements are allowed to have. The HTML Client requires them because it downloads and displays images from HTML Service, the thumbnails of images, by creating in-memory BLOBs and by using **blob://xxxxx** URIs to assign them to elements. It also uses a few data:image URIs in order to display some icons.

font-src

The **data:** value is required. Because it pre-loads fonts to make sure that displayed text is not empty initially before the font is downloaded and suddenly appear when they are.

Custom Response Headers

Info Input Solution comes with an out-of-the-box Apache Tomcat, which hosts 5 web applications. Custom headers can be added to every HTTP response by configuring Tomcat and web applications. To achieve this, a section is added in the WEB-INF/web.xml file for each web application, defining a filter to be utilized. Each filter is then configured using a separate file placed under Info Input Solution/config.

In each of the 5 available Web Applications, under their respective WEB-INF directory, a web.xml file can be found. This *xml* file can be modified with more header filters to strengthen security.

Example configuration for the response header filter

As described above, the configuration files for each filter are placed under Info Input Solution/config. In these *xml* files you define the headers that you want to add to the responses. Here is an example of such a configuration file:

Filter example configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<response-header-mapper>
<!-- generic rule for all html requests -->
<mapping url=".*">
  <default>
    <response-headers>
      <!-- Uncomment the following line to define your custom security policies -->
      <!--header key="Content-Security-Policy" value=""/-->

      <!-- For CL-JAVA allow a 5 minute cache time of all responses. After that time,
      it must revalidate. -->
      <header key="Cache-Control" value="private, max-age=300, must-revalidate"/>

      <header key="X-Frame-Options" value="SAMEORIGIN"/>
      <header key="X-XSS-Protection" value="1; mode=block"/>
      <header key="X-Content-Type-Options" value="nosniff"/>
      <header key="Strict-Transport-Security" value="max-
      age=86400; includeSubDomains"/>
    </response-headers>
  </default>
</mapping>
</response-header-mapper>
```

2.2. Info Input Solution Mobile app

Info Input Solution also offers a mobile scanning application for iOS and Android devices, that enables users to capture high quality images of documents. It is possible to use the device camera to scan physical documents, or import image files that are already stored on the mobile device. The document-associated indexing fields will also be displayed and can be completed before the documents are uploaded to the Core Service. Upon uploading, a new batch is created and will be managed by the Core Service.

2.2.1. Getting started with the Mobile Client application

The *Mobile Client* application is available for download on the AppStore and Google Play store. Any device with a camera can run the app.

At startup, the mobile user needs to provide the *Core Service* URL and port. This is the same information used by the *Thick Client* to start.

2.2.2. Mobile Users Administration

Mobile-enabled *Info Input Solution* systems provide an additional Group permission "Mobile". The system administrator grants this permission to a group, which it then associates with the users that should be able to use the mobile app. There is no limitation to the number of groups and users that have the mobile permission, but at no time can there exist more enabled mobile devices as the system maximum specifies.

The system administrator can view the enabled mobile devices in the [Mobile Licensing Administration](#) screen.

2.2.3. Multi-lingual User Interface

The *Mobile Client* application is available in the following languages:

- English
- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- French
- German
- Japanese
- Polish
- Spanish
- Turkish

At first login, the *HTML Client* application will detect the local operating system language settings and will try to match the user preferred language and display locale settings. After successful login, a user can change the *HTML Client* display language via the *Tools & Options Menu* → *Languages* menu item. The user preference will be stored locally, and future launches of the *HTML Client* will use the latest display language selection.

The *Mobile Client* application will detect and match the language in the current device configuration for language and locale.

3. Jobs Setup

An *Info Input Solution Job* stores the configuration settings to capture documents into *Info Input Solution* and conduct related processing tasks. A *Job* identifies information such as how to name a *Batch*, separate *Batches*, *Folders*, *Documents*, add Indexing information and configure the *Export Destinations*. Before using *Info Input Solution*, you need to create at least one *Job*.

Info Input Solution allows you to define certain objects, like *Document Classes*, *Field Types* or *Scan Profiles* and reuse them in multiple *Jobs*. These objects are called *Shared* and they are:

- [Document Classes](#)
- [Folder Classes](#)
- [Field Types](#)
- [Extraction Profiles](#)
- [Image Enhancement Profiles](#)
- [Data Sources](#)
- [Scan Profiles](#)
- [Export Destinations](#)

To create and modify *Jobs* or any of the *Shared* objects you must have either the *Job Administrator* or *Admin* permission. The *Setup data dialog* is the central point from where you can manage all *Job* related objects. To show the *Setup data dialog*, go to the *Tools & Options Menu* (≡) and select the *Jobs setup...* item:

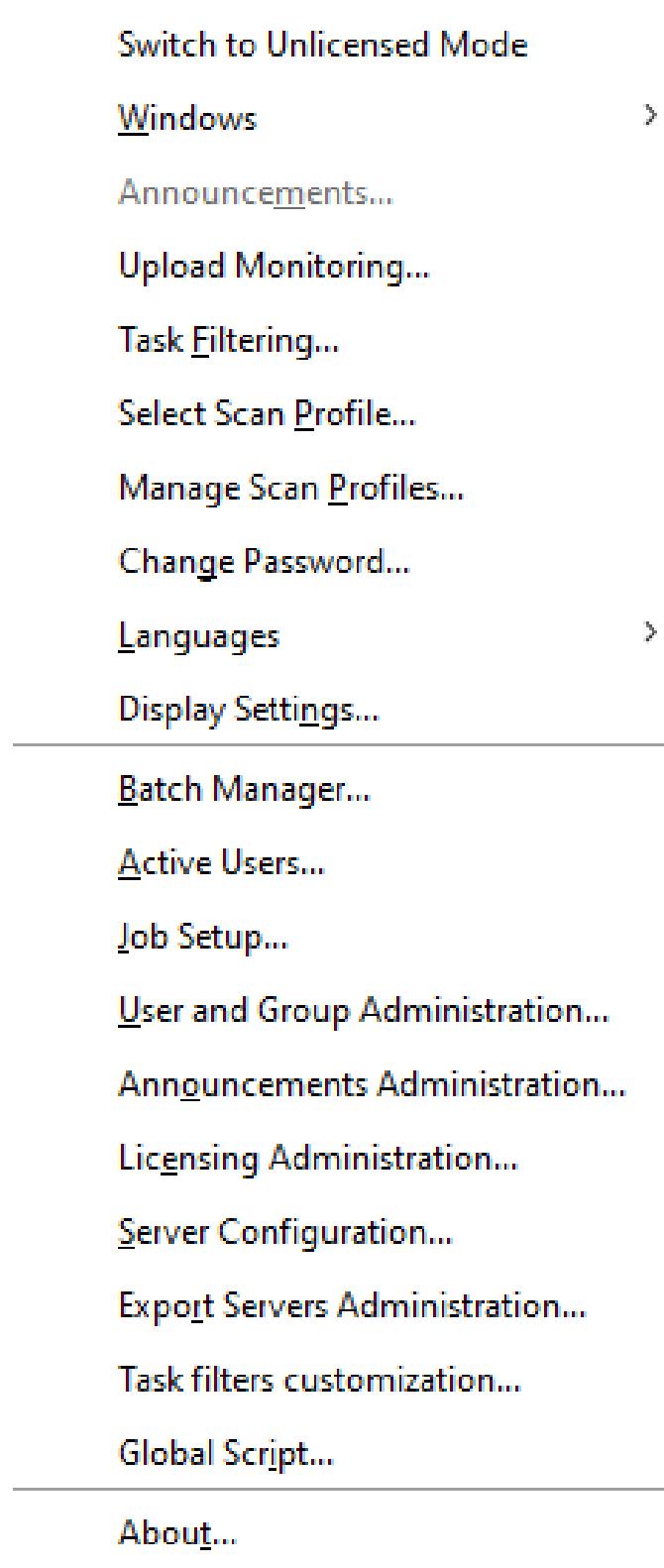


Figure 7. Admin Menu

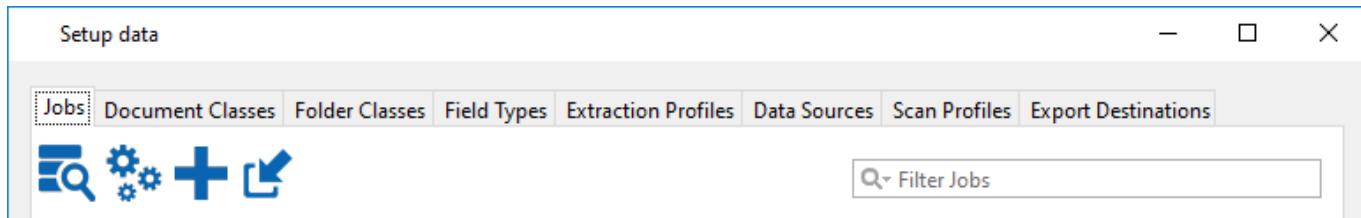


Figure 8. Setup data

The *Setup data dialog* contains nine tabs, one for *Jobs* and one for each type of *Shared* objects. Each tab displays a table with all the available objects of its kind. All the tables have facilities for searching, filtering, sorting and grouping of their data. You may select one or more items in a table and right-click to bring up the context popup menu and perform certain actions.

All the dialogs that are used in *Jobs setup* save their changes to the Core Service when the *Save* button is pressed. In several places, a dialog which edits a certain object may display a second level dialog. In this case the second level dialog will have an *OK/Cancel* button combination instead of *Save/Cancel*. The *OK* button will save the dialog's changes into memory but not to the Core Service. This will allow the user to undo changes made at a second level dialog, by pressing the *Cancel* button at the first level dialog.

All *Shared* objects are checked for concurrent modification. While one (job administrator) user is editing a specific *Shared* object, no other user can modify or delete the same object. If someone tries to edit the same *Shared* object, then a warning message will be displayed at the top of the dialog. Note that, in this case, the dialog operates in read-only mode and the *Save/OK* button is disabled. The second user can still view the properties of the *Shared* object, s/he may even modify some of the values in the dialog, but s/he is not allowed to save any changes.

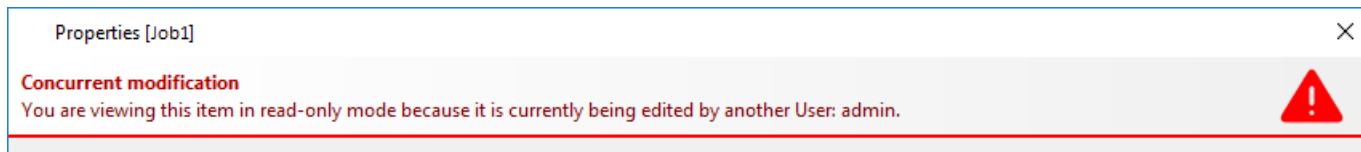


Figure 9. Job properties dialog in read-only mode, when another user is editing the Job.

The *Setup data dialog* is not refreshed with the changes that other users might make. For example, if user-A leaves the *Setup data dialog* open while user-B changes the name of a certain *Shared* object, this change will not appear in user-A's *Setup data dialog*. However, if later user-A attempts to change the name of the same object, then an error will be displayed:

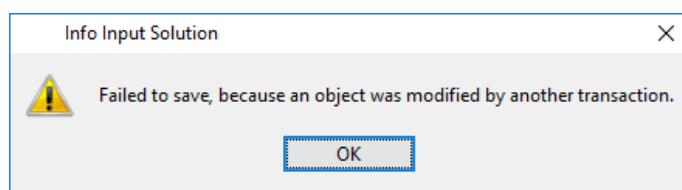


Figure 10. Error while trying to modify version of a Job.

This error indicates that user-A is trying to update a stale version of a *Shared* object. In this case, user-A needs to close the *Setup data dialog* and re-open it in order to update with the latest version of all *Shared* objects.

3.1. Jobs

Jobs can be managed from the *Setup data dialog* at the *Jobs* tab:

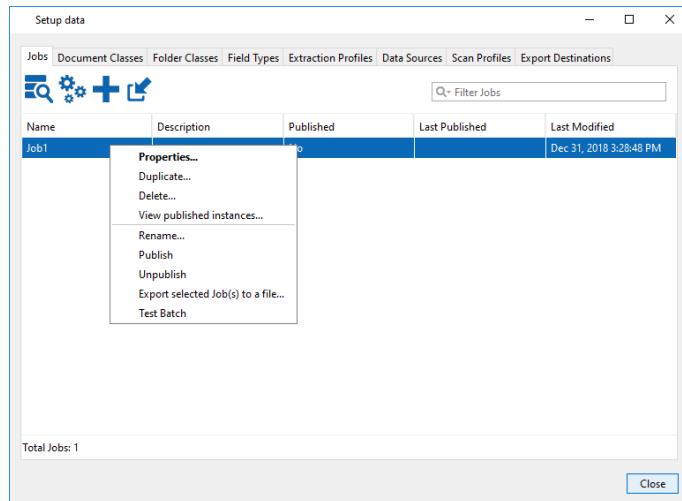


Figure 11. *Setup data dialog: Jobs tab*

The following buttons are available:



(New)
create a new *Job*.



(Import)
import *Jobs* from a file or from a *Kofax Capture System*.

The following actions are available on the selected *Jobs*:

Properties

Open the *Job Properties dialog* to configure the selected *Job*.

Delete

Delete the selected *Job(s)* from the system.

View published instances

Show all previously published instances and publication dates for this *Job*.

Rename

Rename the selected *Job*.

Publish/Unpublish

See below.

Export

Export the selected *Job(s)* into a file.

Test Batch

See below.

3.1.1. Publish/Unpublish Jobs

To *publish* a *Job* is to make it available to the users. A *Job* is not visible/available to users unless it is published. Moreover, every time you update any setting of a *Job*, the new settings do not take effect unless you re-publish the *Job*. The opposite action is to *unpublish* a *Job*, which means that the *Job* is no longer available to users.

When publishing a *Job*, the system actually makes a copy of the *Job* and all associated settings, including any linked/shared objects (like *Document Classes*, *Field Types*, etc,) so that any changes to the *Job* or to the linked objects will not affect the published *Job* from then on. When a user creates a *Batch* based on a *Job*, s/he actually uses the published copy of the *Job* to base the *Batch* on. This copy will never change from then on.

For example, assume that you have a published *Job1* that uses *FieldType1* which contains a list of two possible values. If you add a third value in *FieldType1*, this will be shown to the users only after you publish *Job1*. If you open for indexing any *Batch* that was created before publishing *Job1*, you will still see two possible values in your *Index Field*. This happens because the *Batch* is referencing the published copy of *Job1* before the last change took place. Any *Batch* that is created after publishing *Job1* will show three possible values for your *Index Field*. Note that if *FieldType1* is also used by *Job2*, then you also need to publish *Job2* in order to have this change take effect on any future *Batches* that are created using *Job2*.

It is also important to note that [Datasources](#), [Export Destinations](#) and [Global Scan Profiles](#) are the only *Shared* objects whose changes take effect immediately, even for published *Jobs*. For example, if you change the URL of a *Datasource*, this change will affect all *Batches* from all *Jobs*, even the *Batches* that were created before the *Datasource* change.

3.1.2. Test Batches

Info Input Solution allows you to create a batch for an unpublished *Job* by using the *Test batch* function. In the *Jobs Setup* dialog, select the *Job* you want and click on the *Test batch...* button to create a batch: the batch that will be created will use the *Job* you selected there, even if it is not a published *Job*. Moreover, the copy of the *Job* that will be used is the latest one, (including the latest changes you made) and not the published copy of the *Job*. This function allows you to test-run your batches without publishing your *Jobs*.

Test batches cannot be saved/committed to the Core Service.

Script debugging in test batches

When you create a *test batch*, *Info Input Solution* allows you to *debug* your [Indexing scripts](#) by giving you access to the *Script Editor* in a real-time mode: to do so, create your *test batch*, click on the *Index button* in the main window and then right-click on the *Legend area* of the *Indexing Pane*. The following menu will appear:

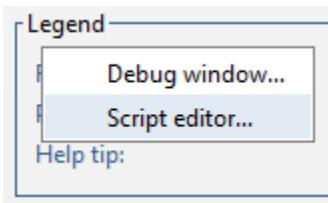


Figure 12. Script debug menu

The *Script editor...* option displays the *Script Dialog*: any change you make to the *Indexing Script* will be compiled immediately and reflected immediately to the *Indexing Pane*. This is the best way to compile and edit your *Indexing Scripts*. See the [Indexing Scripting](#) topics for details. The *Debug window...* displays a window you can use to display debug output.



Figure 13. Debug output window

3.1.3. Import/Export Jobs

Jobs definitions can be imported or exported allowing users to move *Jobs* from one system to another (i.e. from development to production). Only systems running the same version of *Info Input Solution* are supported for moving *Jobs* using the Import/Export function.

Info Input Solution exports *Jobs* definitions to a compressed zip file using the *Export selected Job(s) to a file...* command. You can use the generated zip file to import the *Job(s)* back to another system. You may also use the *Import/Export Jobs* function to make copies of existing *Jobs*. Note that the exported *Job* definition is the current version of each *Job* and not the published *Job* definition. In other words, if you publish a *Job*, make some modification to the *Job* configuration and then export it, then the exported definition will contain the modifications that have not been published yet.

You can import *Jobs* using the *Import Jobs from file...* menu: the *Import Jobs dialog* appears:

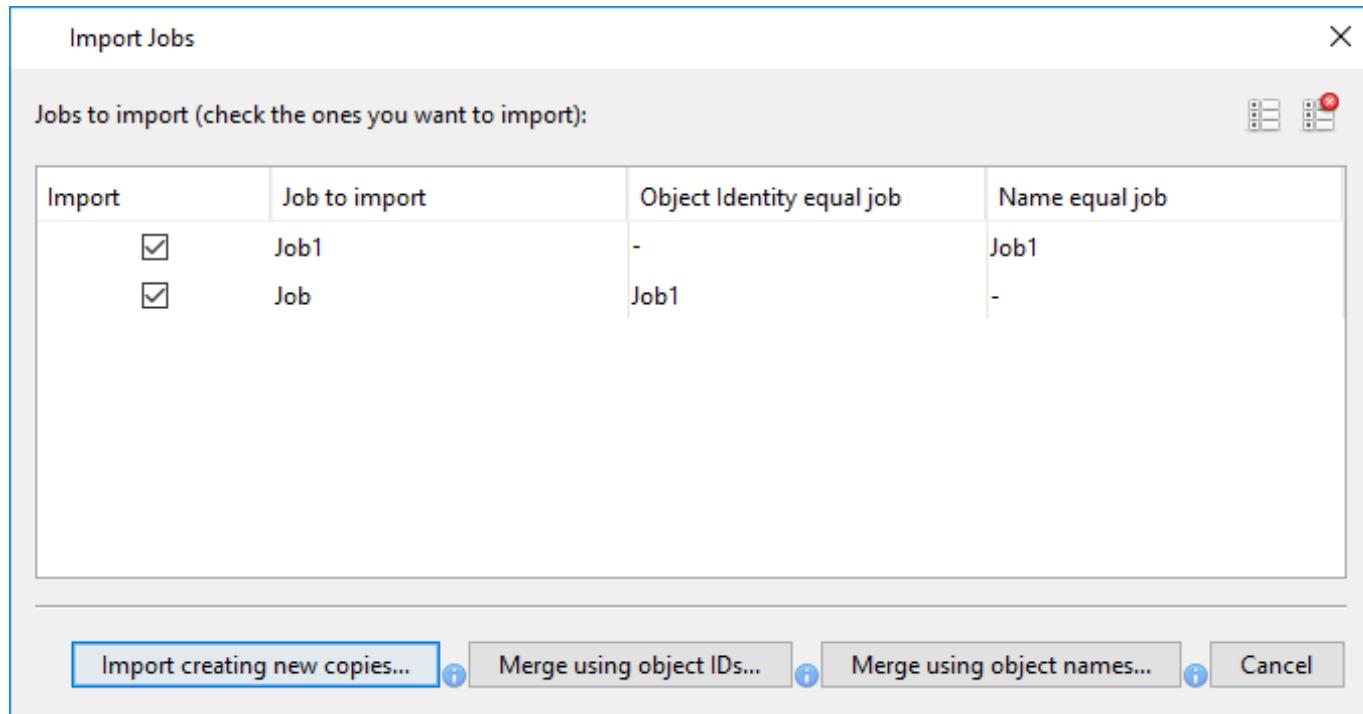


Figure 14. Import jobs dialog

There are three separate modes when importing a *Job*:

Import creating new copies

Using this option, *Info Input Solution* will create copies of all objects when importing.

Merge using object IDs

To match same objects: using this option, *Info Input Solution* will try to match existing objects with the ones in the import file and update their properties, instead of creating new objects, by using their internal object IDs. Object IDs are uniquely assigned to objects when they are first created in a system and they cannot change from then on.

Merge using object names

To match same objects: using this option, *Info Input Solution* will try to match existing objects with the ones in the import file and update their properties, instead of creating new objects, by using their names. Names can be set by the administrator of a system, so they can be used to force the matching of objects and avoid the duplication of entities.

Each mode has its own use case, depending on the reason you are performing the *Import/Export*. For example, if you have a test system where you create *Jobs* and you want to transfer them to a production system, you can select the *Merge using object identities* option, provided that you started with the creation of all objects on the test system. Each object will be created once in the production system, and from then on, its properties will be updated each time you move it from the test to the production

system.

Importing from a Kofax Capture System

Info Input Solution also supports importing *Job* definitions from a *Kofax Capture System* using the *Kofax Import Connector Server*. A new *Job* will be created from a *Kofax batch class*. All the definitions of the batch class are being automatically imported to *Info Input Solution* without requiring any manual configuration. This feature allows a *Kofax batch class* to be linked to an *Info Input Solution Job*. Any batches that are created based on this *Job* will be automatically exported to Kofax through web services for further processing. A *Kofax Export Destination* is automatically created when you Import a *Job* from a *Kofax Server* (see [Kofax Export](#) topic).

3.1.4. Job Configuration

This section describes the configuration steps necessary to configure a new *Job*.

3.1.4.1. General job setup properties

This tab is used to setup some generic options for the *Job*:

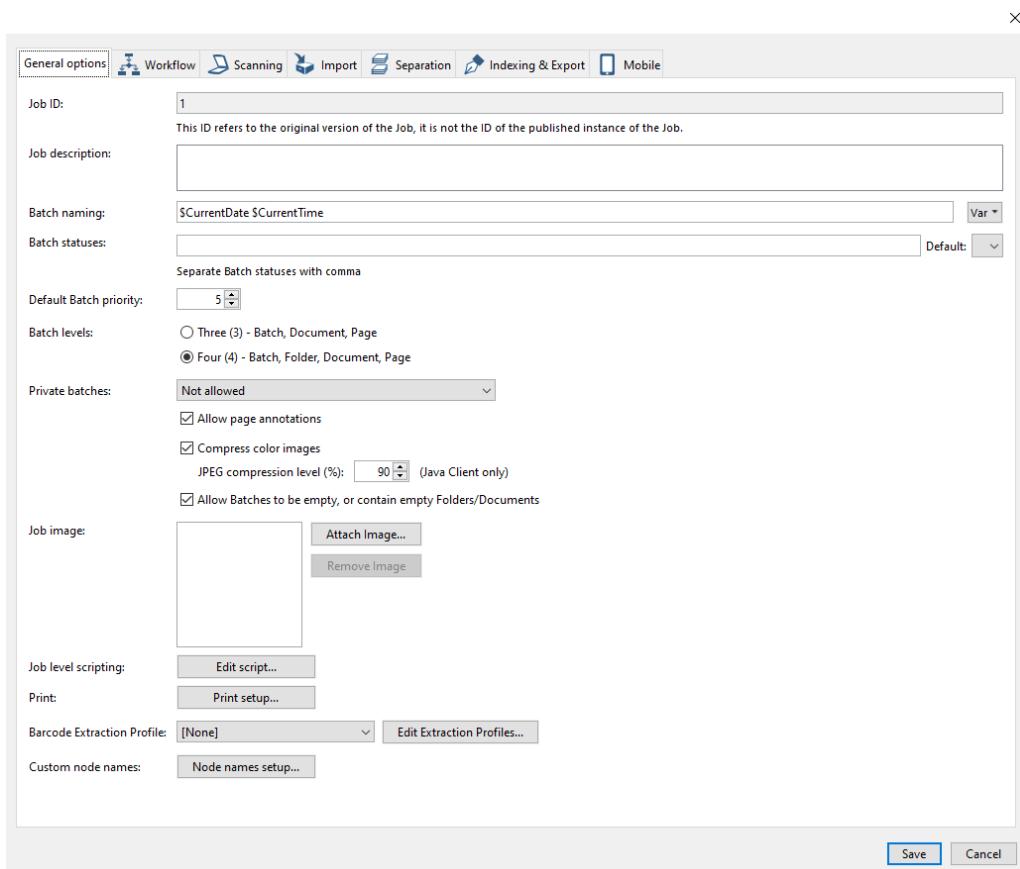


Figure 15. Job properties dialog:General Job setup properties tab

Job ID

This refers to the original version of the Job. It does not refer to the published instance *Job ID*.

Job Description

This is a brief description of the *Job*. This information will be presented to the scanner operators, when they create a new *Batch*, to help them select the appropriate Job to create their batch.

Batch naming

Name to be assigned to the Batch. You may leave this field empty if you want the user to enter a name each time s/he creates a Batch, or you can input a name-pattern that will be used to pre-populate batch names. For example if a job is being defined to capture invoices, then each batch can be named based on the current date and time by entering the following value **Invoice - \$CurrentDate \$CurrentTime**. Notice that the **\$CurrentDate** and **\$CurrentTime** are variables, whose literal values are calculated when the batch is created. There are several *variables* that can be used here: click on the *Var* button on the right to see a list of supported variables. Notice that as you enter an expression in this field, at the bottom of the dialog a *Sample* of the batch name appears: this is done to help you see how the actual batch names will be populated at scan time.

Batch Statuses

This is an arbitrary list of values that can be used as a *status* for each batch. You should enter the values you want separated with comma. The *Default* drop down list is being automatically populated with the list as you type and allows selecting the default value and also checking the distinct status that are recognized. The status of a batch can be selected when creating a batch and can be set at any time by right-clicking on the batch from the *Batch Explorer Tree* → *More options for batch ...* → *Status*. It is also displayed and can be set from the [Batch Manager](#).

Default Batch Priority

Assign a default priority for all batches created for this Job. This is displayed in the batch creation dialog. For example a specific Job may have a higher priority so the administrator can set the default priority to 10 (1 being the lowest priority).

Batch Levels

Select the number of hierarchical levels batches created based on this Job will have. The available options are 3 or 4. Option 3 would be selected to scan batches that have only a document level. Option 4 would be selected to scan batches that also contain a folder level.

Private Batches

This feature will allow you to specify if a batch can only be viewed and worked on by the operator that created it. See [Private Batches](#) topic for more details.

Allow page annotations

allows the users to annotate pages using the annotations tools of *Info Input Solution*. You can disable this functionality by un-checking this option.

Compress color images; JPEG compression level (%)

this option sets whether you want color images to be stored compressed on the Core Service or not. *Info Input Solution* always uses lossless compression for *Black & White* images, yet for color images compression is optional. *JPEG* is a lossy compression format, which means that each time an image is compressed it loses a small part of its quality. For this reason, *Info Input Solution* minimizes the number of decompressions/compressions to the absolutely minimum required. When a color image is scanned and is being viewed and edited in the Client, it remains uncompressed in the temporary cache of the Client: just before the image is sent to the Core Service for the first time, it is compressed (if this option is set), and remains like that on the Core Service. The next time an image is downloaded from a Client to be viewed, it stays compressed in the local cache of the Client, and is not sent to the Core Service again, unless it is changed. So, in a normal work-flow case, where an image is scanned and send to the Core Service but never actually changed after that (e.g. rotated), a color image is only compressed once, no matter how many times a batch is opened and closed. Remember that color images are very large in size and sending/storing them uncompressed may cause network delays.

Allow Batches to be empty or, contain empty Folders/Documents

If this option is not checked the Batches will not be allowed to close or suspend if there are any empty Documents or empty Folders in the Batch.

Job Image

Use this option to add an Image to the specific Job. This image will be shown in the new batch dialog so the end users can select a job setup easily.

Job level scripting → Edit script...

Display the *JavaScript editor* dialog that allows you to write a *Job level script*.

Print → Print setup...

displays the Print dialog in a special mode that allows you to setup several options concerning the printing of batches.

Barcode Extraction Profile

Select the Barcode Extraction profile that is to be used during file import from the drop down list. The button *Edit extraction profiles* is a shortcut to the [Extraction Profiles](#), from this window it is possible to add new Extraction profiles. Note, only barcode extraction profiles will be available in the drop down list.

Custom Node Names

From this menu it is possible to configure custom node names dynamically. For example, it is possible to set the names of the Folders / Documents / Pages according the name of a Folder class of according to a variable value.

Note that the job name is limited to 60 chars, the batch name is limited to 255 chars, and the batch description field is limited to 1,000 chars.

Note that it is possible to use only lowercase letters in custom variables for *Custom Node Names*.

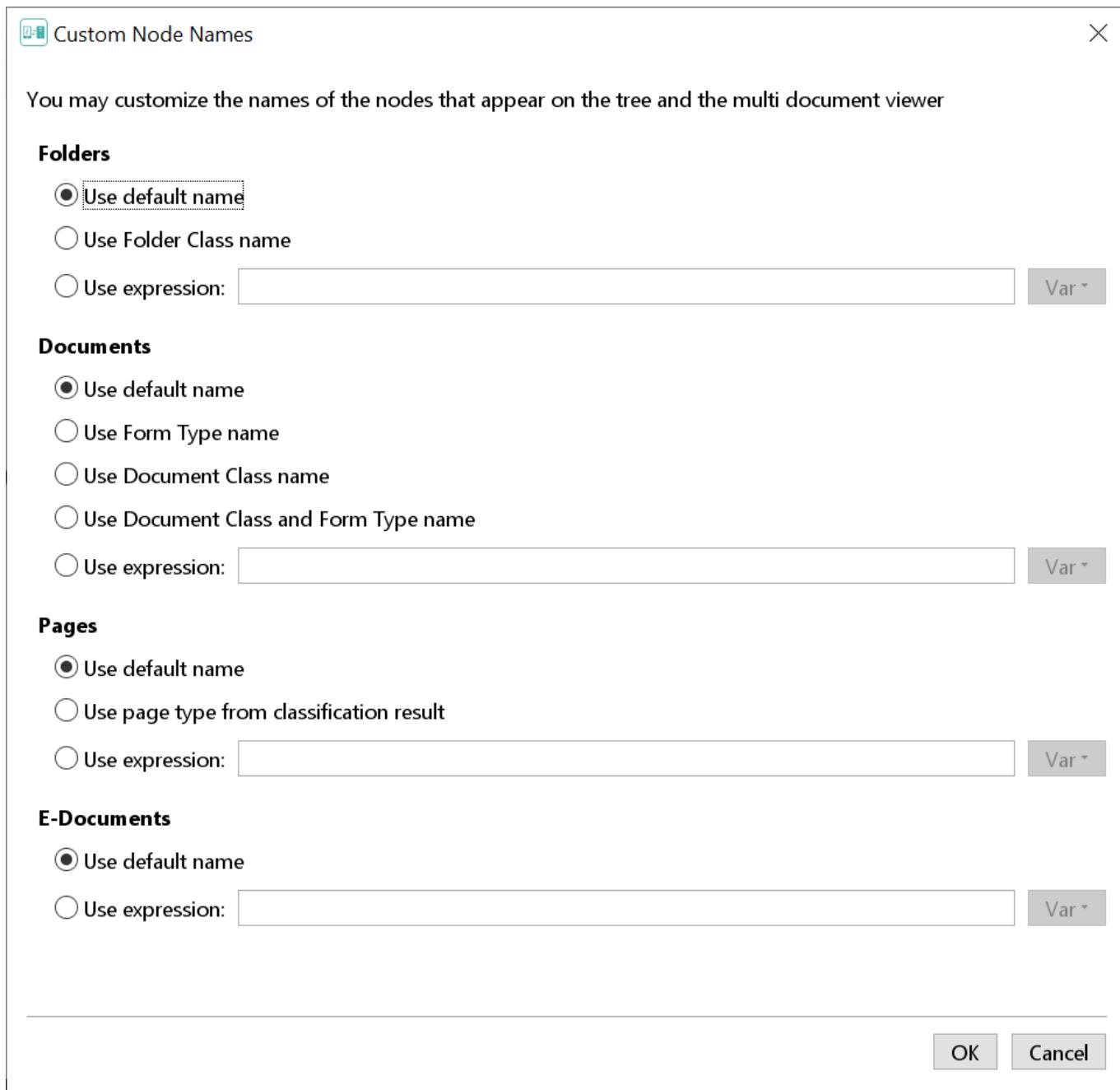


Figure 16. Custom node names

3.1.5. Workflow

A *Workflow* is a high-level process followed by an organization to accomplish a certain goal. Each *Workflow* consists of several processing steps that are executed in series, either by an operator or from the Core Service. A document capture workflow is able to work both with human users, performing difficult, delicate processing on documents, and with automated systems, performing unattended massive processing.

The job administrator can define the workflow process that should be followed by every batch, created from this Job setup. The scenario (Scan, Index, Export) is used by default for every new Job.

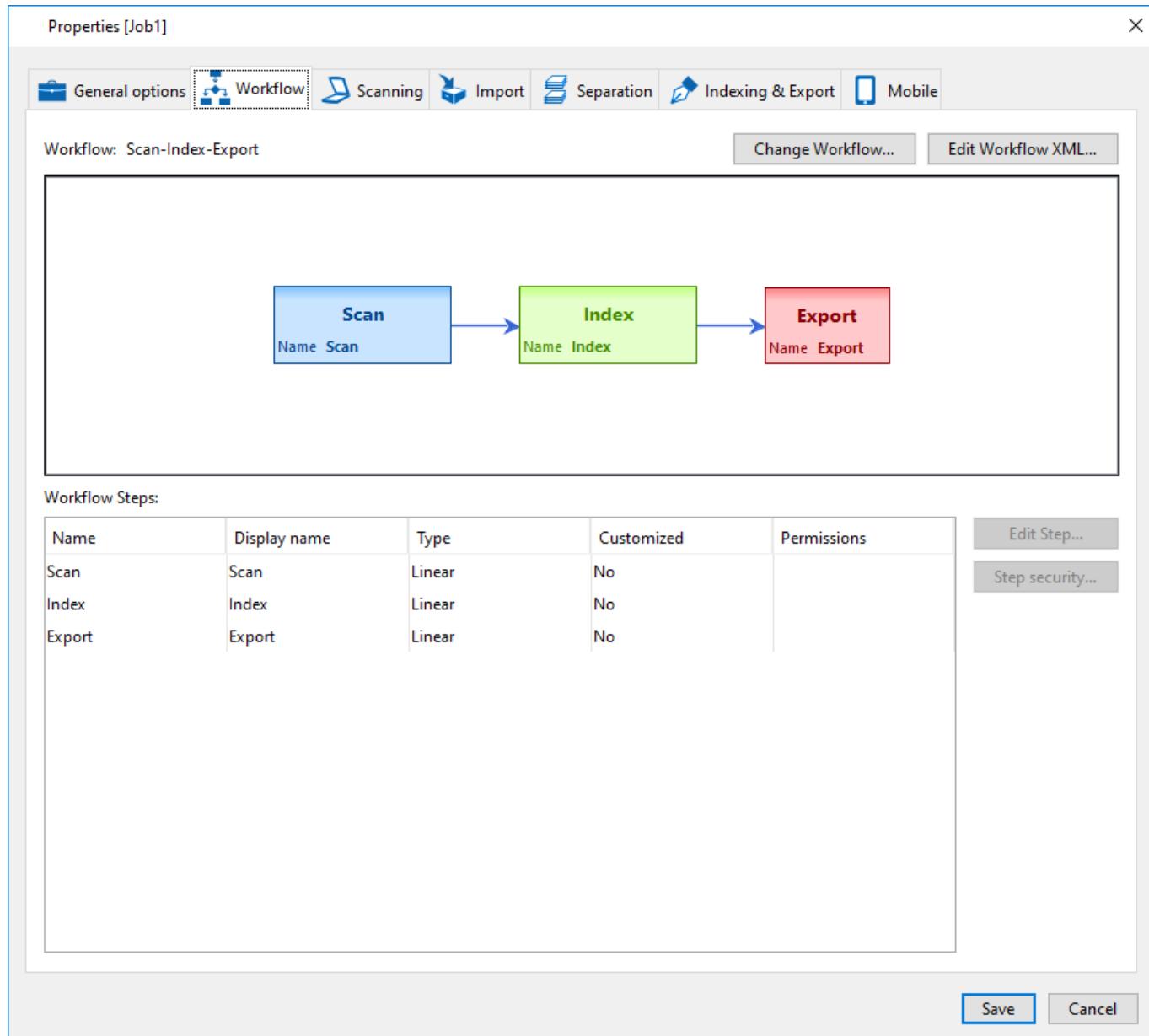


Figure 17. Job Properties dialog: Workflow tab

Change Workflow: On the *Workflow* tab you can view and select the Job Workflow from the predefined list of supported *Workflow* scenarios.

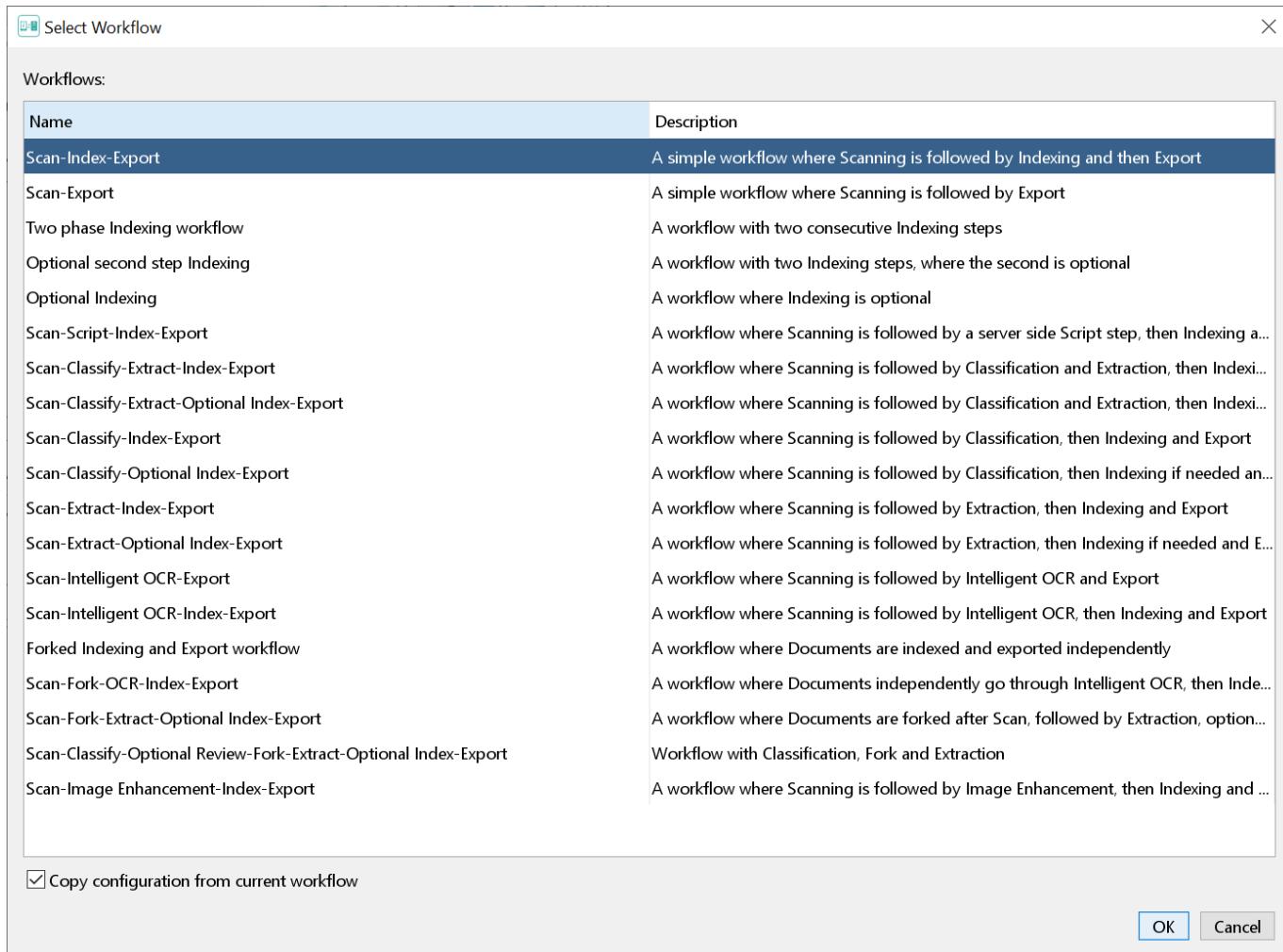


Figure 18. Select Workflow

Edit Workflow XML: It is also possible to design custom Workflow scenarios, by editing and renaming one of the available default scenarios. More details on [how to edit the Workflow XML](#). === Print setup

Info Input Solution allows users to *print* a batch to a local or network printer. Several options regarding the printing of a batch can be setup on the *Job* level, much like creating a printing template for the batch. All options regarding batch printing can be allowed or disallowed for the user, and default values can be provided in the form of static or dynamic expressions.

The *Print dialog* in setup mode is as follows:

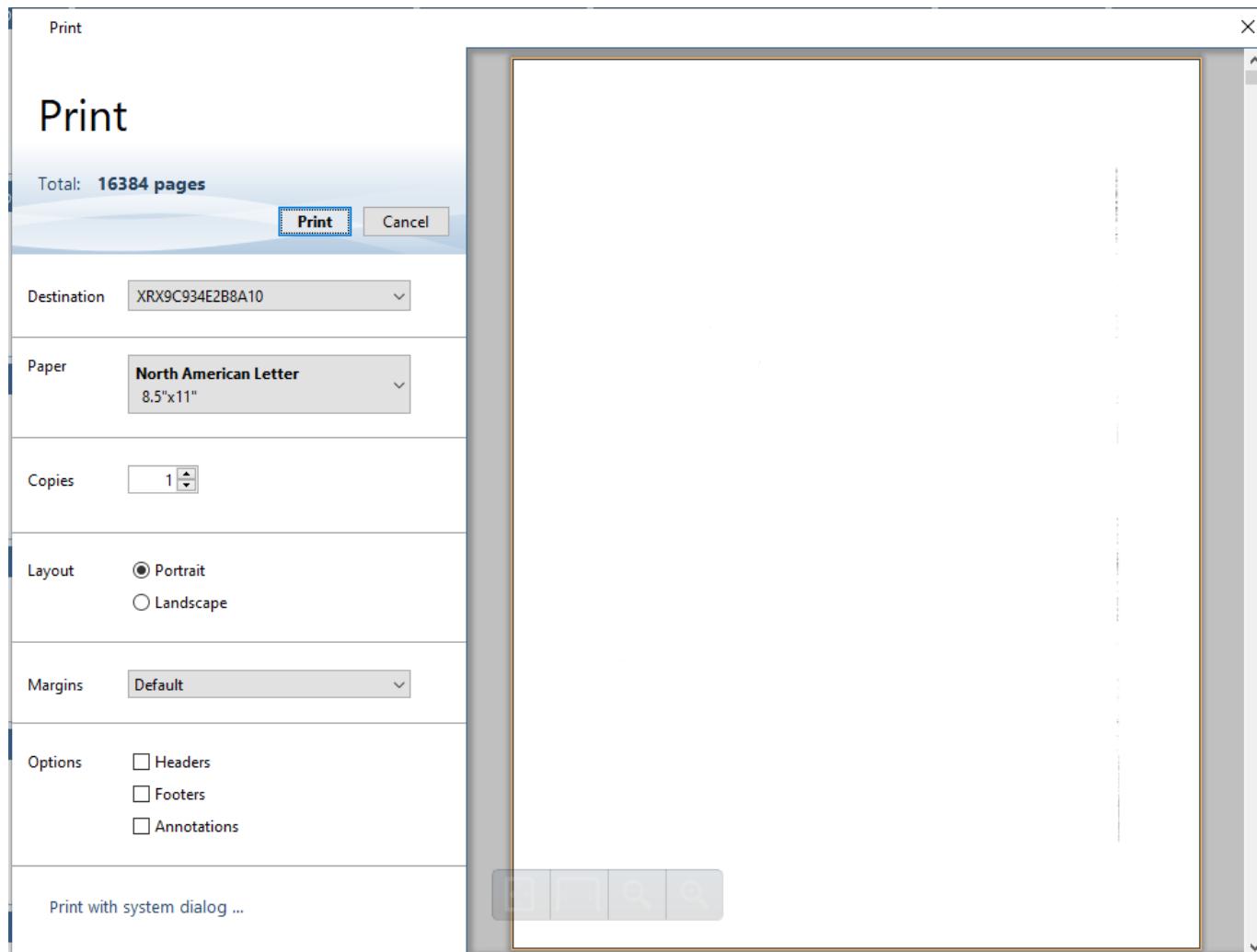


Figure 19. Print dialog in setup mode

Info Input Solution can overlay an optional *header* and *footer* at each page when printing. There are three areas defined in the header and footer, where specific information can be overlaid: left, middle and right. The *Print Setup* dialog has six edit boxes, 3 above the *Sample page* and 3 below the *Sample page*, where you can insert predefined dynamic expressions that are evaluated at print time.

During setup, you can insert in either of these 6 text boxes an expression which will then be evaluated and overlaid when printing. There are several default variables you can use like *total number of pages*, *current page*, etc. The following mini-toolbar provides shortcuts to these variables:



Figure 20. Quick variables toolbar

You first need to click in one of the text-boxes (left, middle, right), and then click on the appropriate button to insert the variable you want in the dynamic expression. For example, if you want to print the cur-

rent page at the bottom of each page, you can write in the bottom-middle text box:

Page \${printPageNumber} of \${printPageCount}

Moreover, you can select the *font* for the header and footer from the corresponding *font selectors* at the top and bottom of the *Sample page* area, and also choose whether you want a separator line between the header/footer and the page content. Notice that as you change settings in this dialog, the *Sample page* is updated in real time to display a sample of the print result. You may also select whether you want to print a *header, footer* (in total) and *annotations* or not, from the option selectors on the left area.

It is important to note that all the options in this dialog are used as default values for the user. Moreover, as an administrator, you may choose whether the user can override them or not. For each option you set, there is a corresponding checkbox in the *End user may* list on the left, where you can decide whether the values you set can be altered by the user at print time. For example, if you want to disallow users to print annotations, you need to uncheck the *Annotations* option and also uncheck the *End user may → enable/disable annotations* option. The same logic applies to all other options.

For example, if you want to print the full path "*Batch name / Document name / Page name*" at the top of each page printed; you can put at the header (3rd text box at the top-right):

```
 ${parent.parent.name} / ${parent.name} / ${name} or ${batch.name} / ${document.name} /  
 ${name}
```

If you don't want users to be able to change that header, you should also uncheck the *End user may → edit right header*.

3.1.6. Scanning

From this tab you may assign specific *Global Scan Profiles* and *Scan Properties* to a *Job*, in order to better control the properties of the scanned documents.

3.1.6.1. Controlling Scan Profiles for a Job

Each Job can be associated with a number of [Global Scan Profiles](#):

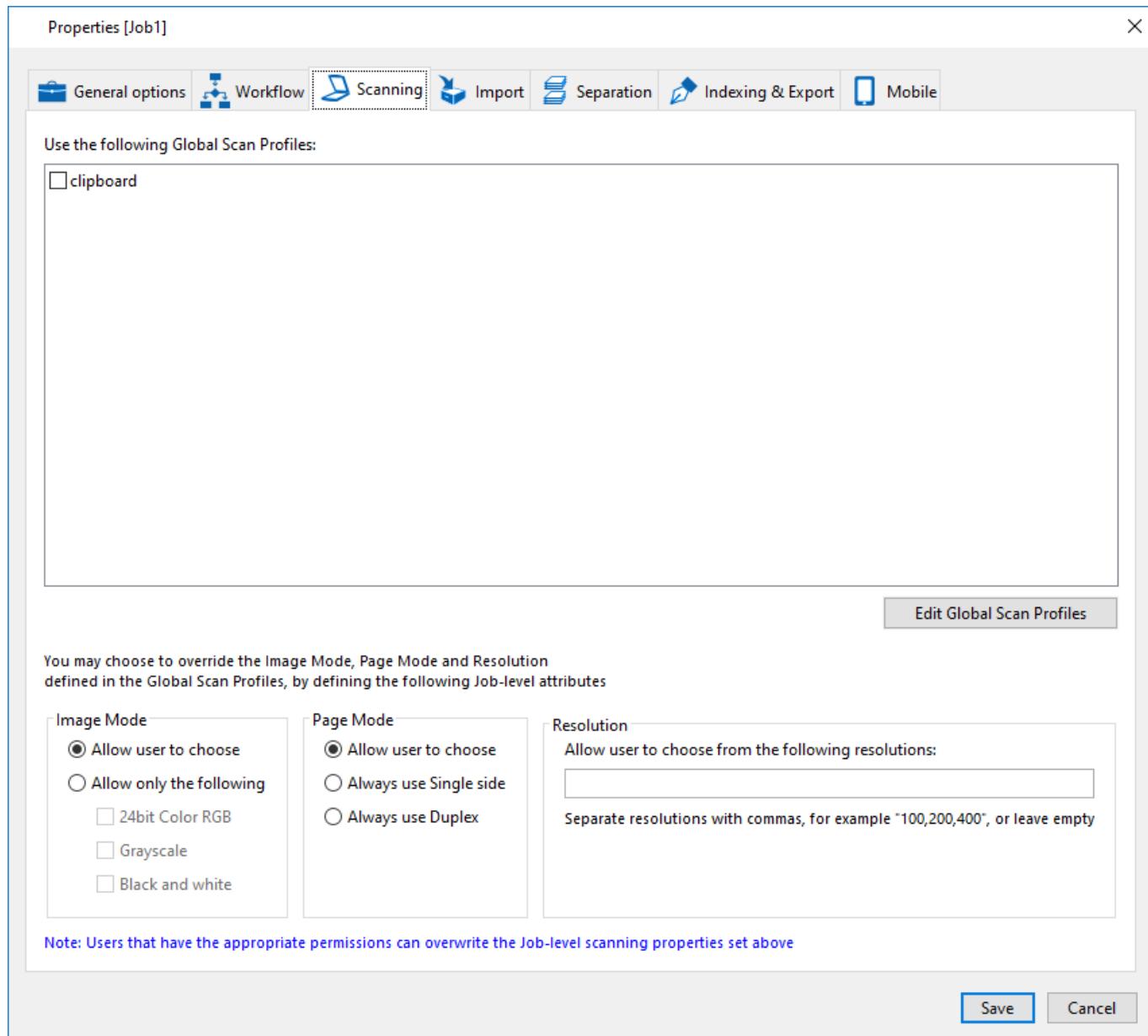


Figure 21. Setup Scan Profiles and Properties for a Job

On the *Global Scan Profiles* list, select the *Scan Profiles* that you want to assign to the *Job*. When a user creates a Batch from that Job, s/he will be forced to select one of the allowed *Scan Profiles*. If a *Job* has no *Scan Profiles* assigned, then users are able to select one of their *local Scan Profiles*.

Note that Users who have the *Overwrite job scan profile* permission will be allowed to use their *local Scan Profiles* in addition to the *Global Scan Profiles* assigned to the Job.

3.1.6.2. Controlling Scan Image Mode for a Job

You can control the *Image Mode* used when scanning for a specific Job. The available options are *Color*,

Grayscale, Black and White or any combination of those. To impose specific *Image Modes* for a *Job*, click on the *Allow only the following* radio button and select the desired *Image Modes*. When a user scans documents for this *Job*, the *Image Mode* selected here, will override the setting that was chosen during the *Scan Profile* creation through the scanner's native dialog.

If you select *Allow user to choose*, then users will be allowed to use any of the three *Image Modes*.

3.1.6.3. Controlling Scan Page Mode for a Job

You can control the *Page Mode* used when scanning for a specific *Job*. The available options are *Single Side* and *Duplex scanning*. To impose a specific *Page Mode* for a *Job*, select the appropriate radio button on the *Page Mode* sub-panel. When a user scans documents for this *Job*, the *Page Mode* selected here, will override the setting that was chosen during the *Scan Profile* creation through the scanner's native dialog.

If you select *Allow user to choose*, then users will be allowed to use either *Single Side* or *Duplex scanning*.

Note that users who have the *Overwrite job scan page mode permission*, will be allowed to choose between the two *Page Modes*, regardless of the value assigned on the *Job*.

3.1.6.4. Controlling Scan Resolution for a Job

You can control the *Resolution* used when scanning for a specific *Job*. You may assign more than one *resolution* to a *Job*, by entering a *comma separated list* of allowed values. *Resolution* values are given in *dpi* and should be between **10** and **20000**. Please note that *Info Input Solution* does not check whether the values entered here are actually supported by the scanner's driver. If you enter a resolution value that is not supported, it will result in an error at scan time.

When a user scans documents for this *Job*, the *resolution* values selected here will override the setting that was chosen during the *Scan Profile* creation through the scanner's native dialog.

If you leave the *Resolution* textbox empty, then users will be allowed to select the *resolution* they desire when scanning.

Note that users who have the *Overwrite job scan resolution permission*, will be allowed to choose the scan resolution, regardless of the value assigned on the *Job*.

3.1.6.5. Use Cases

1. *You want to give full control to users when scanning:* In this case, the *Job* is not associated with any *Global Scan Profile* and no *Image Mode*, *Page Mode* or *Resolution* is specified. Users will be allowed to use any of their *local Scan Profiles*. They will also be allowed to determine which *Image Mode*, *Page Mode* and *Resolution* to use.

2. *You want to allow two different scanner models for a specific Job and want to allow full control over the scanning properties:* In this case, you can create *Global Scan Profiles* for each scanner model, configure them using the scanner's native dialogs and assign them to the *Job*. This allows you to control several properties and functions for each scanner, for example paper source, paper size, de-skew, auto-crop etc. In addition, you can specify the *Image Mode*, *Page Mode* and *Resolution* for this *Job*. Users are obliged to choose between the two available *Scan Profiles* and they cannot alter any of the scan properties. Typically each user will only have to select the *Scan Profile* that is appropriate for the scanner model s/he is planning to use.
3. *You are only interested in restricting the Image Mode:* In this case, you do not need to create and assign *Global Scan Profiles*. However, you want to make sure that everything is scanned in *Black and White* mode so you need to assign this *Image Mode* to the *Job*. Users will be allowed to use their *local Scan Profiles*, but the *Image Mode* will be forced to *Black and White*, even if this is not what was selected when the *local Scan Profile* was created.

3.1.7. Import Options

On the *Import* tab, you can setup some import options for the *Job*. You may configure if a new batch will require indexing and/or if a scanning operator will be allowed to import non image files (eDocuments) such as MS Office documents or videos. There are five sections in this dialog that allows you to customize how the system behaves when you import (either by *drag-n-drop* or by using a *Folder* type [Scan Profile](#)).

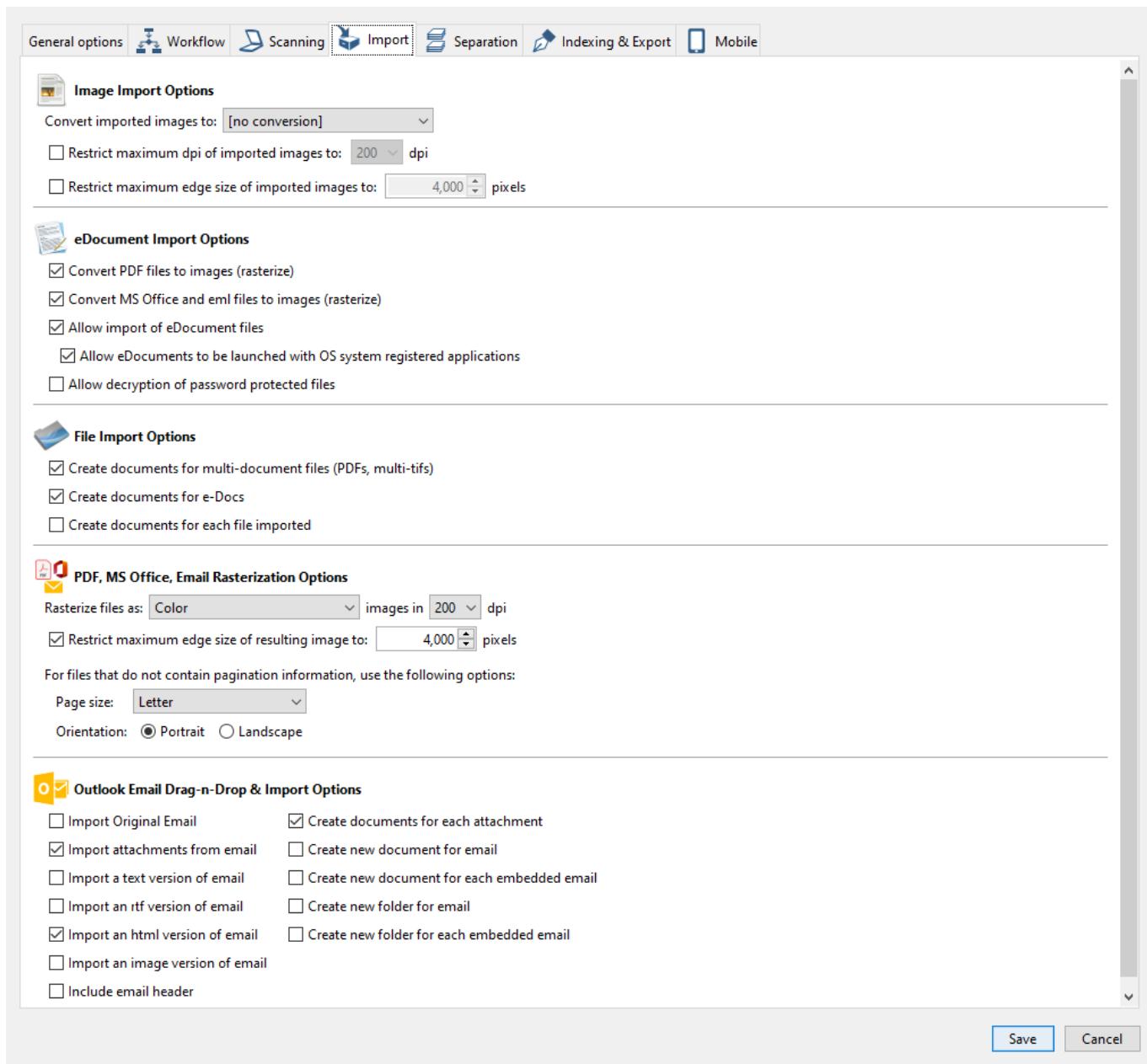


Figure 22. Import Options tab

3.1.7.1. Image Import Options (includes image-based PDF files)

Convert imported images to

normally, you would not want images in PDF pages to be converted to a different color depth. But you can choose from this option to convert images in a specific color depth. Notice that this option does not refer to any image in the PDF file, but only to those PDF pages that are used as a container to store scanned images. Note that, this option affects all types of imported images as well as PDFs that are used Image Containers

Restrict maximum dpi of imported images to X dpi

if this option is selected, the maximum dpi analysis will be restricted.

Restrict maximum edge size of imported images to X pixels

if this option is selected, the maximum edge (width or height) is restricted to this number of pixels.

3.1.7.2. eDocument Import options

Convert PDF files to images (rasterize)

Info Input Solution will take apart PDF files and convert them to image files (TIFF) as it imports them, if you don't select this option, PDF document files are imported as they are (e.g. as eDocument files).

Convert MS Office files to images (rasterize)

Info Input Solution will take apart Ms Office files and convert them to image files (TIFF) as it imports them, if you don't select this option, MS Office document files are imported as they are (e.g. as eDocument files).

Allow import of eDocument files

option enables the import of electronic document files.

Allow eDocuments to be launched with OS system registered applications

sets whether you want the native registered application to be launched to display an eDocument. For example, if a user imports an MS Word file, then MS Word will be launched to display this file, if the user double-clicks on it to display it. There is a security concern here since eDocuments include executable files that may harm the computer they are executed on. For example, a malicious user may add an executable file as an eDocument which, when run, harms the target PC. There are three different ways to protect the target system from such attacks:

- You can deselect this option, so users cannot view/display/execute any eDocument (they can only import them): this is the most restrictive approach since it also cuts off much of the power of using eDocuments.
- You can use the *Job level scripting* method `checkFileAllowed` to only allow files with specific extensions to be imported. Using this method you can exclude any files (such as `.exe`, `.bat`, etc) that may cause harm when executed. This is similar to what other programs, such as MS Outlook, do with attachments that are considered harmful.
- You can use the *Job level scripting* method `batchCreated()` to select which eDocuments you want to display inline instead of launching the registered OS application/viewer. For 'dangerous' files (like `.bat` files) you may want to display them as text with the embedded viewer, instead of actually launching them on the target PC.
- You can use the Import Options dialog to further customize the behavior of the system when importing files (see below).

- *Allow decryption of password protected files* enables the automatic detection and decryption of encrypted documents, like password protected PDF files and Microsoft Office files. *Info Input Solution* is capable of detecting password protected files and will offer to the user the ability to decrypt them. When this functionality is not enabled at the Job level, then password protected files will be imported as eDocuments and the user will not be able to decrypt them.
- By default, *Info Input Solution* supports decryption for password protected PDF files. *Info Input Solution* can also handle password protected Microsoft Office files.

The following table summarizes the supported password protected Microsoft Office files:

File Type	Extension
Word	.docx, .docm
Excel	.xls, .xlsx, .xlsm
PowerPoint	.ppt, .pptx

Info Input Solution is also able to support other types of password protected Microsoft Office files, provided that the corresponding Microsoft Office application is installed on the workstation. More specifically:

- If Microsoft Word is installed on the workstation, then the Thick Client will be able to handle password protected files with .doc extension.
- If Microsoft Excel is installed on the workstation, then the Thick Client will be able to handle password protected files with .xlsm extension.

3.1.7.3. File Import Options

Create documents for multi-document files (PDFs, multi-TIFFs)

selecting this option will create a new document for each multi-page file being imported. This applies to *PDF* and *multi-TIFF* files. There is a special provision for single-image TIFF files whereas they are treated as multi-page files if the TIFF tag **TIFFTAG_PAGENUMBER (297)** is present.

Create documents for e-Docs

this option will create a separate document for each *non-image file* being imported, no matter its type.

Create document for each file imported

this option will create a separate document for each file imported, *image or non-image*. For example, if this option is checked and you import 5 TIFF files, the system will create 5 different documents, each containing one of the imported images.

3.1.7.4. PDF Rasterization Options

A PDF file consists of one or more PDF pages. The following options apply if the *Treat imported PDF files as images* option has been selected. In that case, *Info Input Solution* will take apart the PDF file and create a separate image from each *PDF page*. *Info Input Solution* treats *PDF pages* in two distinct ways:

- If the *PDF page* contains a single image and no text or annotations, then *Info Input Solution* will extract this single image from the container *PDF page* and consider it to be the full page. The dpi value of the resulting image is calculated using the pixel size of the image inside the PDF and the PDF page size.
- In all other cases, the *full PDF page* is rasterized internally and the output is then converted to an image file (internally it is converted to a TIFF file) - these pages are considered as *text PDF files*.

Notice that a single PDF file may contain pages that fall in either of the above cases: each PDF page is treated separately (although a single PDF file usually contains one type of pages).

Rasterize text PDF files as <IMAGE COLOR MODE> images in X dpi

the color mode option specifies the color depth of the rasterized images. The dpi value represents the detail (or zoom level) that the PDF page is rendered. A value of 96dpi corresponds to a zoom value of 100% when you view the PDF from a viewer like Acrobat Reader. For files that will be OCRed, a minimum value of 200dpi is suggested.

Restrict maximum edge size of resulting text PDF image to X pixels

if this option is selected, the maximum edge (width or height) is restricted to this number of pixels. This option makes sense in order to avoid very large images that may result if you try to import a PDF file with very large physical dimensions.

3.1.7.5. Outlook Email Drag-n-Drop & Import Options

This set of options applies to *Outlook email messages*. *Info Input Solution* supports dragging email messages directly from *Microsoft Outlook* or the import of *.msg* files, which is the format that *Outlook* messages are stored in when on disk. In that case, *Info Input Solution* will take apart the message and any attachments it may contain (including other attached messages, with infinite recursion) and create appropriate documents. The first set of options specify the format that the body of an email will be imported:

Import original email (.msg file)

if this option is selected, a copy of the whole *.msg* file is imported, as an *eDocument*. Use this option if you want to preserve the email message in the native Microsoft Outlook format.

Import attachments from email

if this option is selected, then any attachment of the email is imported as separate *eDocument*.

Notice that Outlook messages may contain images and other objects inline the message body: these are treated as attachments. Special provision exists for signature images which are discarded.

Import a text version of email

if this option is selected, a plain-text version of the email is imported: notice that any formatting and non-text elements (like inline images) are lost.

Import an rtf version of email

if this option is selected, then an rtf version of the email is imported, if available

Import an html version of email

if this option is selected then an html version of the email is imported, if available. Outlook usually stores email as html or rtf, so if you check both these options then you will end up with the actual formatted email body.

Include email header

The header of the Email will be included in the imported files.

The following three options specify how attachments of emails are grouped in documents and folders when imported:

Create documents for each attachment

if this option is selected, then *Info Input Solution* will create separate documents for each attachment.

Create new Document for email

if this option is selected, and the *Job* supports it, then a new Document will be created to hold all pages that are created during the import of a single email.

Create new Document for each embedded email

since Outlook emails may contain other outlook emails as attachments, this option will create a separate Document for each embedded email.

Create new folder for email

if this option is selected, and the *Job* supports it, then a new folder will be created to hold all documents that are created during the import of a single email.

Create new folder for each embedded email

since Outlook emails may contain other outlook emails as attachments, this option will create a separate folder for each embedded email.

When an Outlook email is imported, the system creates dynamic properties (e.g. variables that can be accessed using the `${property.<property name>}` syntax or the `node.properties['<property_name>']` syntax from Javascript) for the email body node that contain the properties of the email. Notice that only those properties with valid values are created. (The properties creation depends of the imported email content. For example, there is a possibility that after an email import the property `displayTo` will not be created but the `recipient.N.displayName` will be created).

Specifically, the available properties are:

Variable name	Explanation
<code>import</code>	a standard property set to <code>true</code> for each imported message.
<code>name</code>	display name
<code>namePrefix</code>	display name prefix
<code>displayTo</code>	display To field
<code>displayCc</code>	display CC field
<code>displayBcc</code>	display BCC field
<code>priority</code>	message priority (e.g. <code>normal</code>)
<code>receivedByEmailAddress</code>	received by address (this is specific to the mail server, e.g. for an Exchange server this is usually the full DN attribute of the internal LDAP directory)
<code>receivedByName</code>	full name of the user of the mailbox
<code>recipient.N.type</code>	this attribute (along with the next two) are repeated for each recipient (N=1, 2, 3, ...). The type is <code>T0</code> , <code>CC</code> , <code>BCC</code> , etc.
<code>recipient.N.displayName</code>	the display name of the Nth recipient
<code>recipient.N.smtpAddress</code>	the smtpAddress of the Nth recipient
<code>replyTo</code>	the reply-to address of the email message
<code>senderAddressType</code>	the sender's address type (e.g. <code>EX</code> for Exchange)
<code>senderEmailAddress</code>	the sender's address (the format depends on the type)
<code>senderName</code>	the name of the sender
<code>subject</code>	email's subject
<code>subjectPrefix</code>	email's subject prefix (e.g. <code>FWD:</code> , <code>RE:</code> , ...)
<code>categories</code>	any categories this email belong to

Variable name	Explanation
clientSubmitTime	date and time the sender submitted the message
commonEndTime	end date and time of the message
commonStartTime	start date and time of the message
creationTime	date and time the message was created
flagDueBy	date and time specifying the date by which an e-mail message is due
flagIcon	flag icon of the message object
flagStatus	flag state of the message object
guid	message's global unique id
importance	the message sender's opinion of the importance of a message
internetMessageId	unique ID for the message
keywords	keywords of message
lastModificationTime	date and time the message was last modified
messageDeliveryTime	date and time the message was delivered
hasAttachment	true, if the message has attachments

3.1.8. Separation

The separation feature defines the rules that will be used to automatically separate documents, folders and batches during scan time. Separators can be a *barcode* on a page, a *patch code* on a page, a *blank page*, a specified *number* of pages or documents and a *batch separator sheet*. When using a separation rule, *Info Input Solution* will create a new document, folder or batch (depending on the configuration) when the separator rule is detected by the system.

3.1.8.1. How are separation rules applied

Info Input Solution supports an arbitrary number of separation rules that are applied sequentially after each page is scanned or imported in a batch. The *Separation* tab displays the list of separation rules that are currently active for this Job:

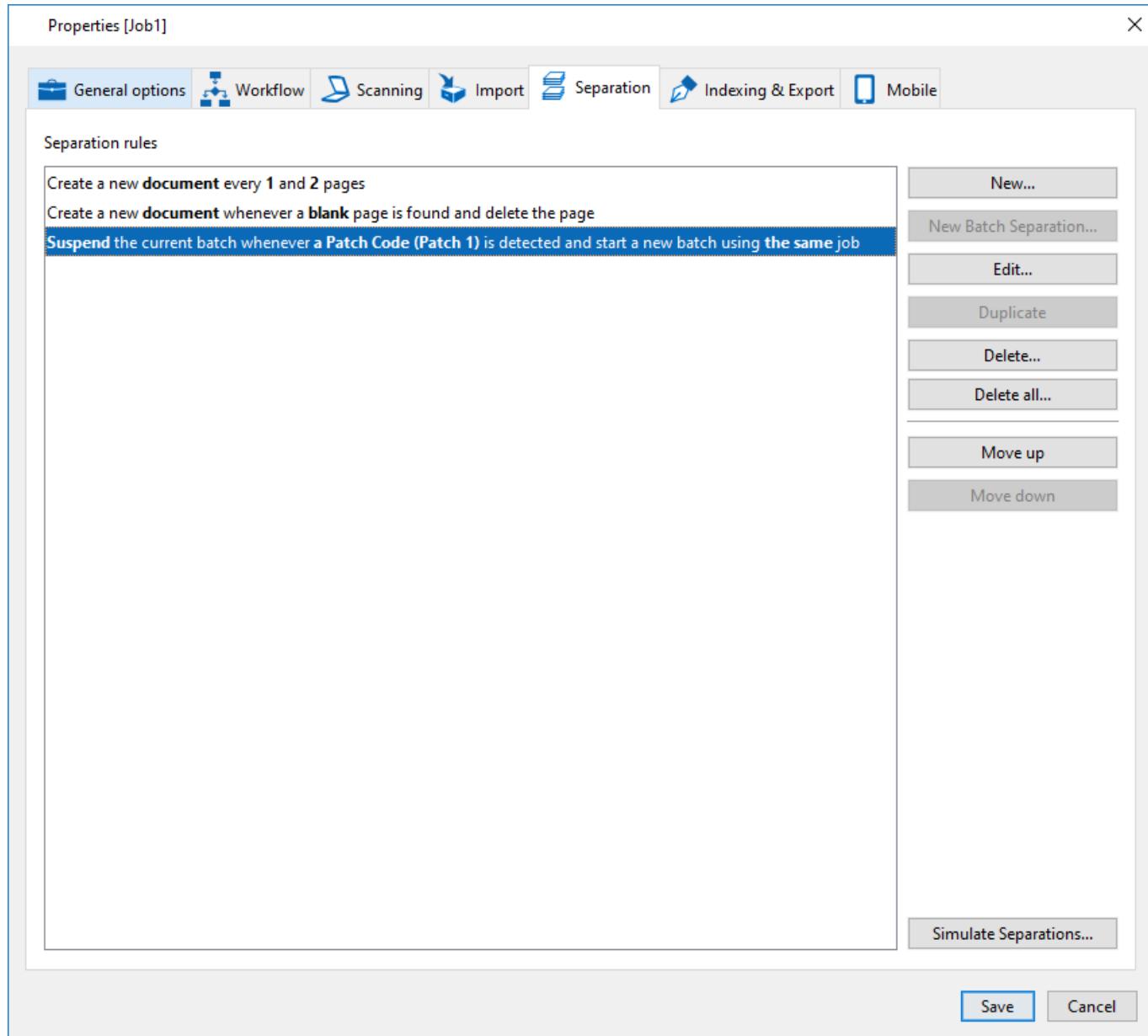


Figure 23. Separation rules list

Since separation rules can get complex and even inconsistent if they are set up without caution, there is a facility to test them without actually scanning any pages: you can click on the *Simulate Separations...* button at the bottom of the *Separation* tab to display the *Test separation rules* dialog:

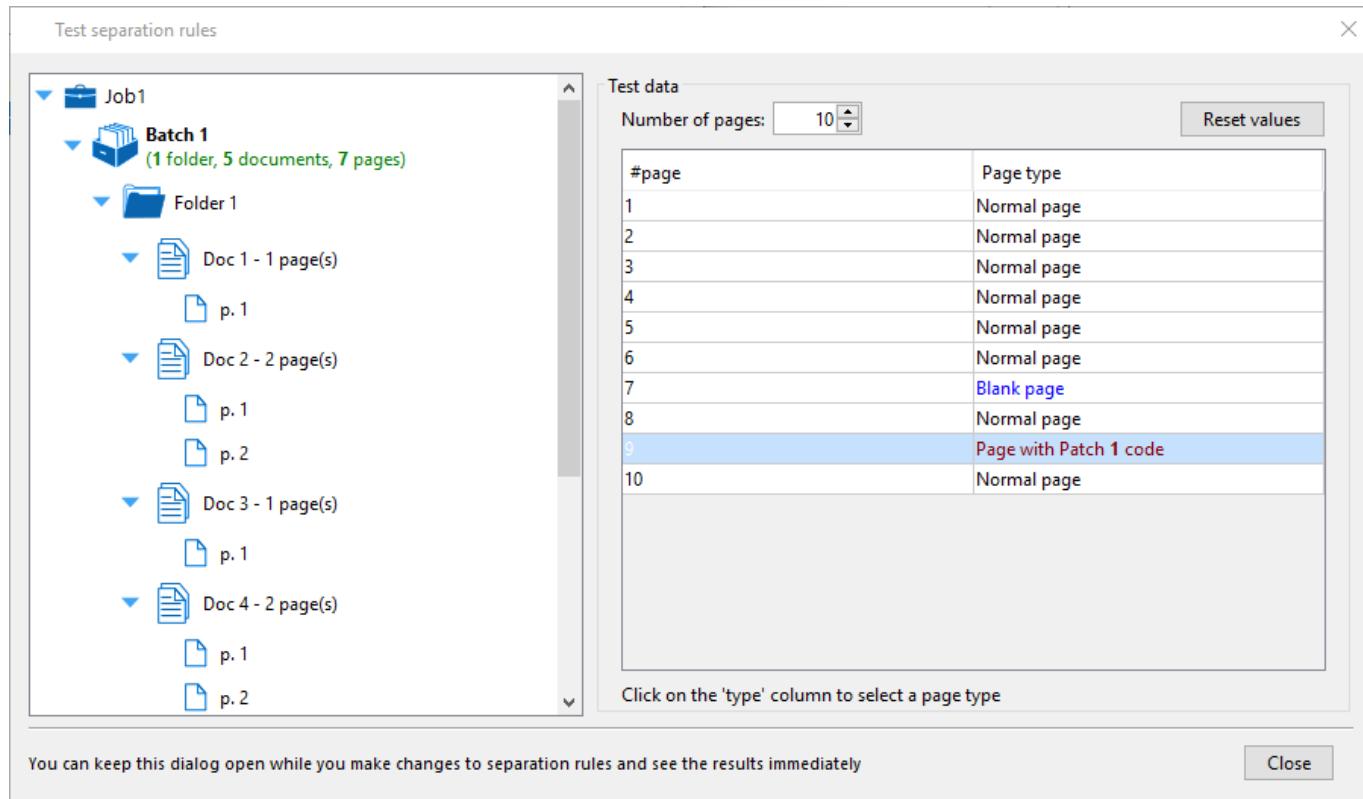


Figure 24. Test separation rules dialog

From this dialog you may click on the *Page type* column and alter the type of each page to emulate what type of page is being scanned: as you do so, the *Batch tree* on the left is updated according to the current separation rules to display the structure as it would be if the separation rules would have been applied:

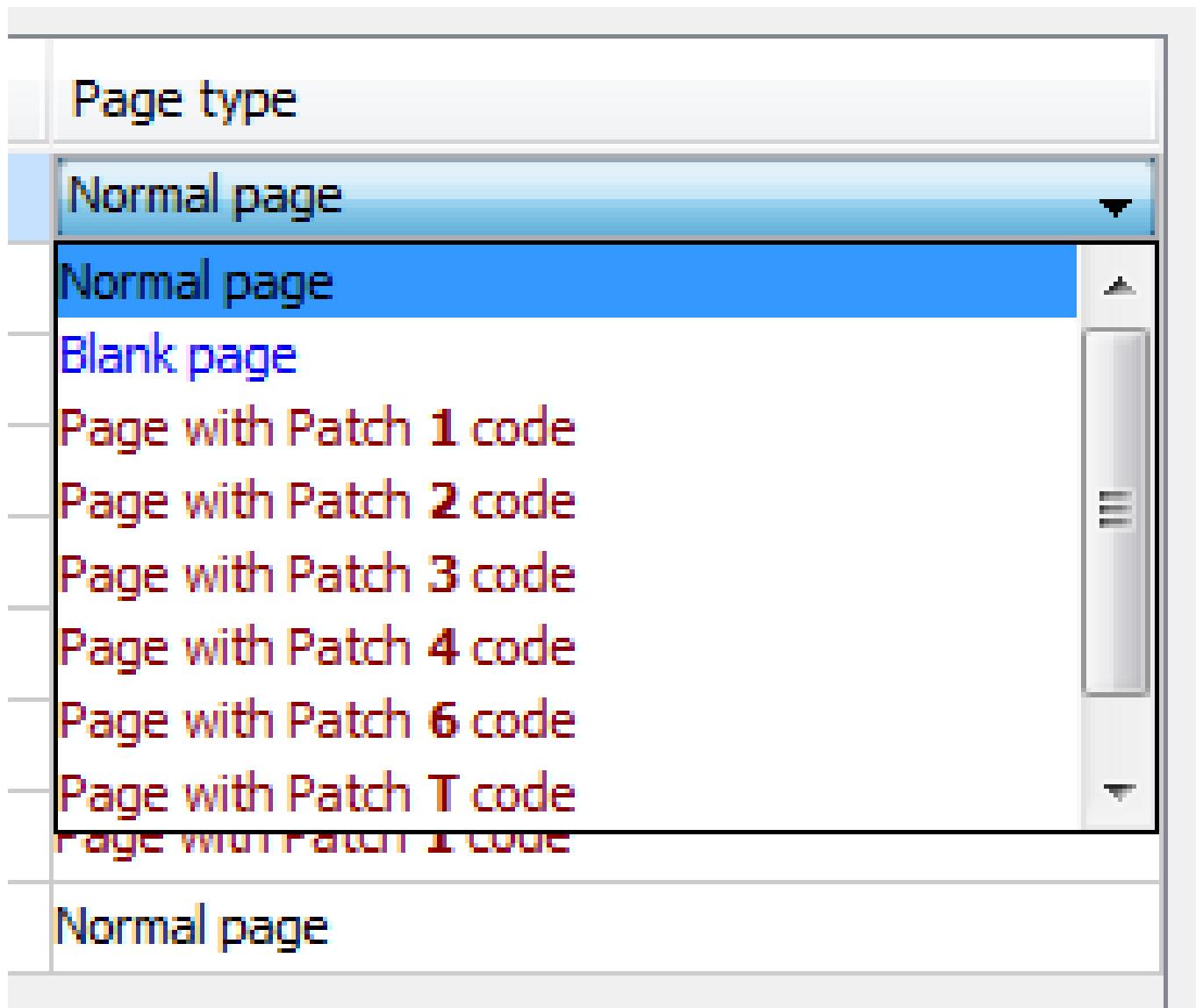


Figure 25. Changing Page type in the Test Separation rule dialog

You may thus test-run the separation rules to see how they will be applied at scan-time. You may also increase the *Number of pages* if you need to emulate a larger batch of pages.

3.1.8.2. Creating/Editing separation rules

The steps below describe how to setup a separation rule;

- *Step 1:* Select an action for when separation should occur.

Step 1: when does this action take place?

Action: For every x-y-z-... pages

For every x-y-z-... pages

For every x-y-z-... images

For every x-y-z-... documents

When a blank page is detected

When a Patch code is detected on a page

When a Bar code is detected on a page

Step 2: how often

For every x-y-z pages

Step 3: define the options of the event

Action	Explanation
For every x-y-z pages	Trigger event after a specific number(s) of pages.
For every x-y-z images	Trigger event after a specific number(s) of images. Each side of a double sided page is considered a separate image.
For every x-y-z documents	Trigger event after a specific number(s) of documents.
When a blank page is detected	Trigger event when a blank page is detected.
When a Patch code is detected on a page	Trigger event when a Patch Code is detected on the page.
When a Bar code is detected on a page	Trigger event when a Bar Code is detected on the page.

- *Step 2:* Enter how often the event in *Step 1* occurs before separation. In case *For every x-y-z pages* was selected then more than one values can be entered, separated by comma. For example if the entered value in the *For Every* text box is 1,2 that means that the system will separate on the first and then on the third page. That action will be re-triggered for the rest of the batch in the same sequence.
- *Step 3:* Define the options for the event selected in *Step 1*. *Options* tabs will be enabled based on the action selected in *Step 1*:
 - *Blank page options* are enabled if *When a blank page is detected* is selected in *Step 1*: Enter the threshold to be used to determine if a page should be considered blank:

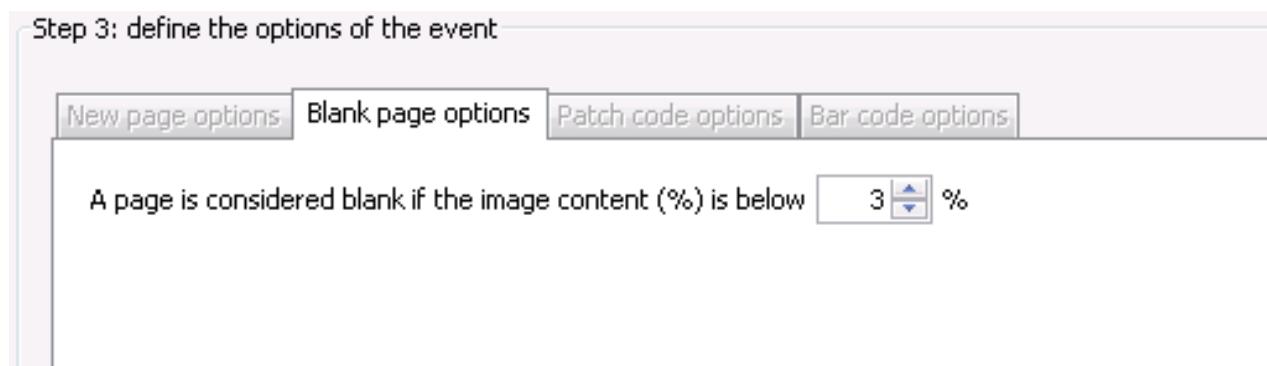


Figure 26. Blank page options

- ° *Patch code options* are enabled if *When a patch code is detected on a page* is selected in Step 1: Select the appropriate *Patch Code* that the system should detect. For example a *Patch 2* patch code can be used to separate documents and another separation rule can be defined to use *Patch 3* as a folder separator. Note that if a patch code is used in a *Batch separation rule*, then it cannot be used in a Document/Folder separation rule.

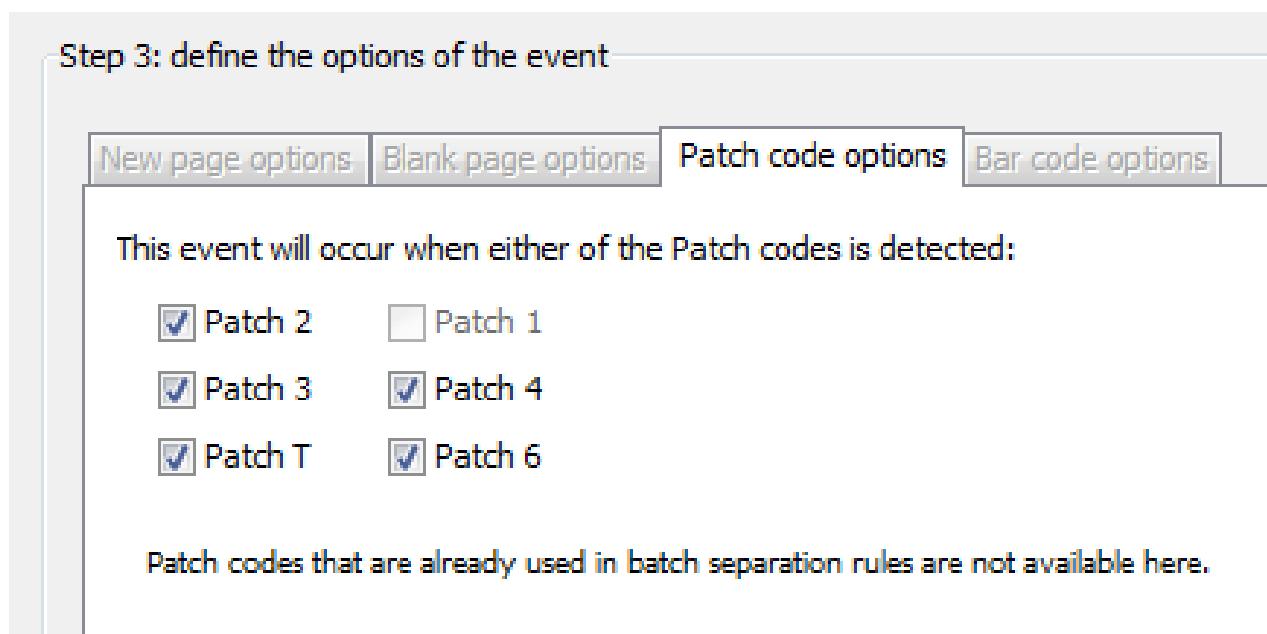


Figure 27. Patch code options

- ° *Barcode options* are enabled if *When a Bar code is detected on a page* is selected in Step 1: select the appropriate barcode to detect on the page. In order to avoid detecting specific barcodes on a page, the *The bar code text must match the following regular expression* option can be checked to enable the text box below to enter a regular expression: as a result only the bar codes, whose text match the regular expression, will be considered for the separation.

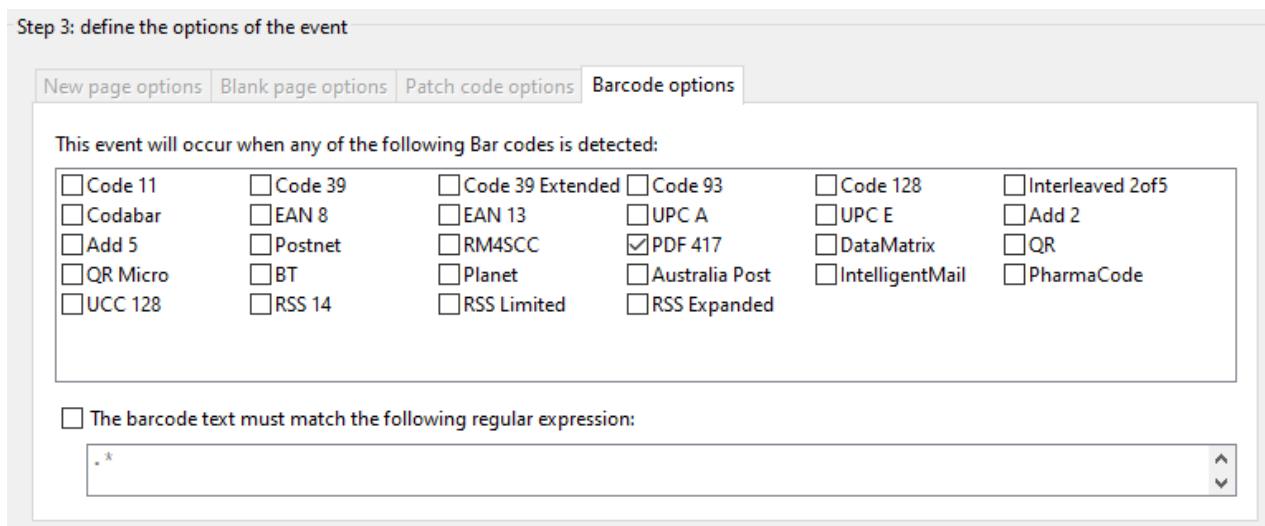


Figure 28. Bar code options

- *Step 4:* you may define what action should be taken if all the rules in steps 1-3 apply. The available options in this step depend on the number of hierarchical levels of the Job (3 or 4). In case of 4 levels, then the *create a new Folder* option is also available. The option to delete the page is available by checking the *and then delete this page*.

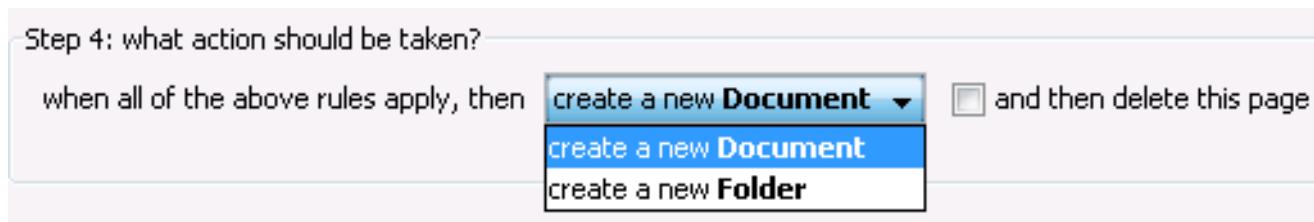


Figure 29. Action to be taken

3.1.8.3. Creating/Editing Batch separation rules

Batch separation rules can be used to automatically create new Batches while scanning. The Batch separation rule is triggered when the appropriate Patch code is detected on a page. The current Batch can be either *suspended* or *closed*. In addition, the Job to use for the next Batch can be the same as the Job of the current Batch, or it can be encoded on the separator page. Note that, Batch separation rules are only available in the Thick Client.

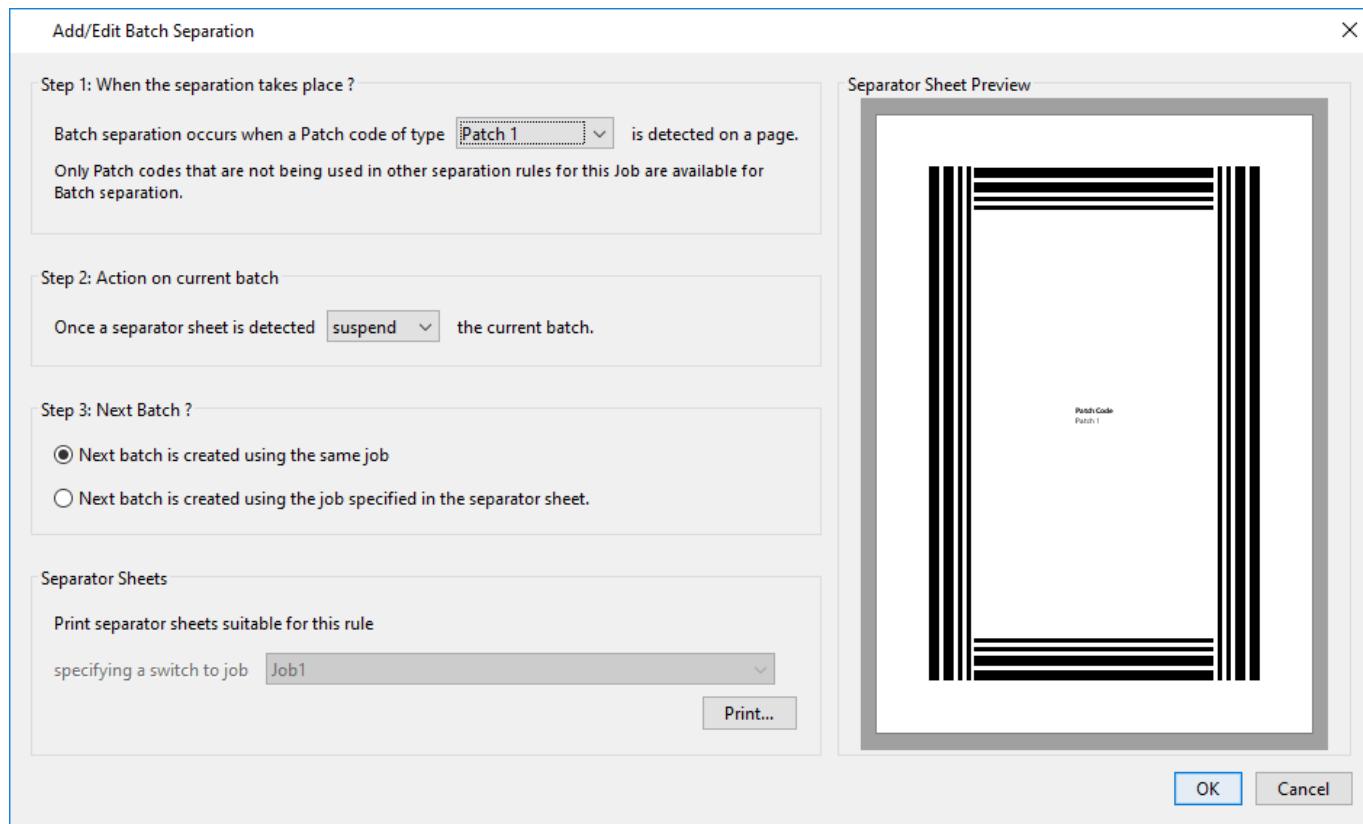


Figure 30. Batch separation dialog

Note that Patch codes which are used in other separation rules, will not be available for use in a Batch separation rule.

The Batch separation dialog provides the ability to print separator sheets. Each separator sheet has the selected Patch code along the four edges. In addition, if you choose to have the next Batch use the Job specified in the separator sheet, then the sheet includes 4 barcodes with the name of the selected Job.

When the Job for the next Batch is specified in the separator sheet, it is possible that the next Job will require specific Scan settings (Scan Profile, image mode, page mode or resolution) which will conflict with the ones being used while scanning. In this case, the next Batch cannot be automatically created and scanning will be aborted:



Figure 31. Batch creation fails due to conflict

3.1.8.4. Customizing the separation logic

The separation logic can also be customized using job-level scripting, and specifically the `pageArrived()` method: The job administrator should refer to the *Info Input Solution Developer's Guide* for the java script template and comments, for use case examples. Note that, when more than one barcodes (or patch codes) exist on a separation page, the scripting functionality can process them in a sorted left-to-right and top-to-bottom order. Moreover, the script-based separation functionality can also be used to detect and delete blank images or two-sided pages, during Scan or Import page.

The logic defined in job-level scripting will be evaluated before the Job specific Separation rule list, defined in *Job Properties* → *Separation* tab.

3.1.9. Indexing and Export

On the *Indexing and Export tab* you define *Batch level indexing* properties, as well as which *Folder* and *Document Classes* this *Job* uses.

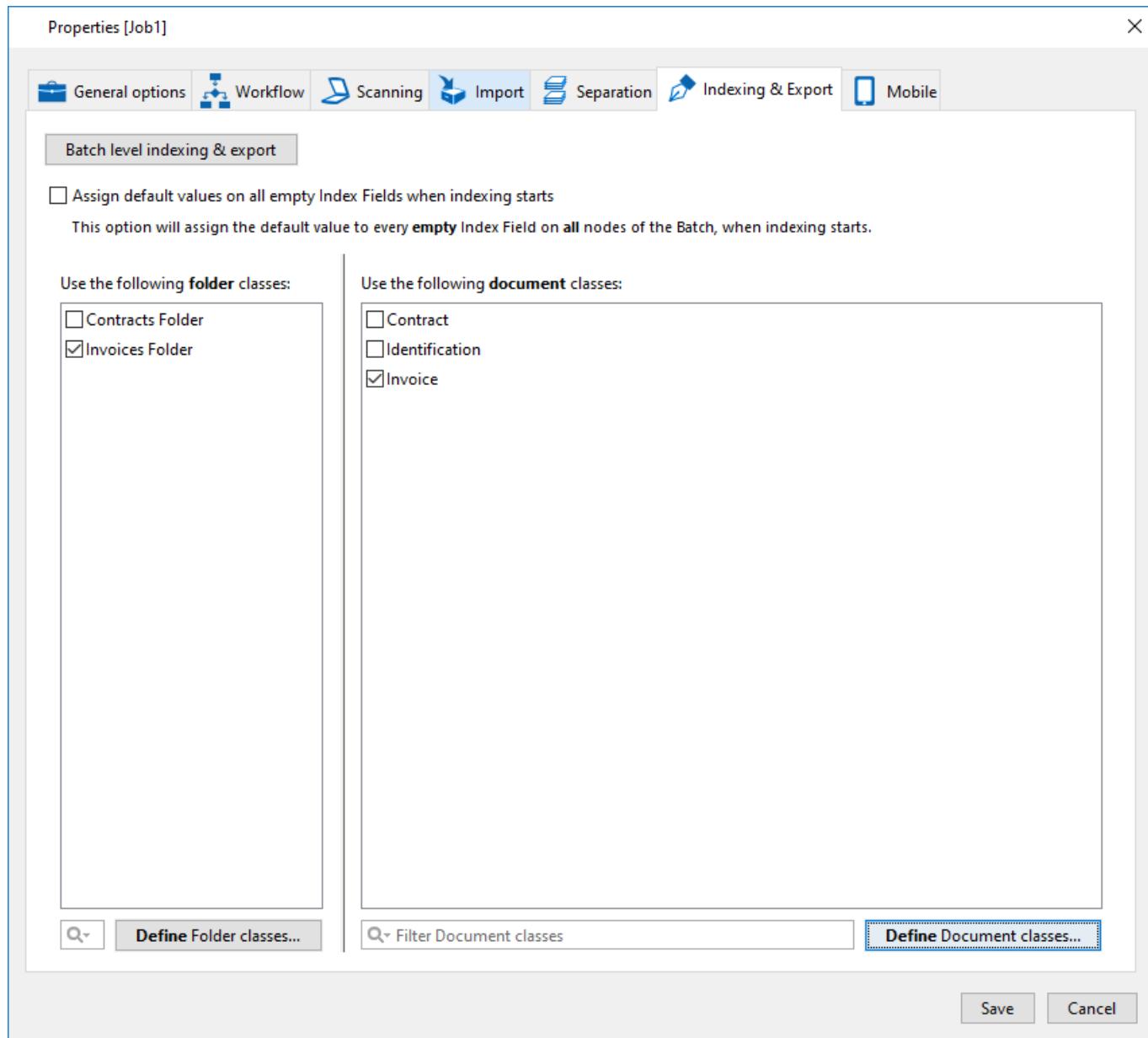


Figure 32. Job properties dialog: Indexes & Export tab

This tab also displays two lists with all available *Folder* and *Document Classes*. Check the *Folder* or *Document Classes* that you want to assign to the *Job*.

During indexing, the index operator is actually choosing *Form Types* to associate with documents, not *Document Classes*. Since you, as an administrator, associate one or more *Document Classes* with a *Job*, *Info Input Solution* takes all the *Form Types* of all the *Document Classes* that are associated with a *Job* and presents them as a unified list to the index operator to choose from, during indexing. When the index operator selects the desired *Form Type*, the *Document Class* is indirectly being selected, the one where the *Form Type* belongs to, and thus *Info Input Solution* presents then the proper *Index Fields* to the index operator to proceed. Notice that all *Form Types* that belong to the same *Document Class* share the same

index fields.

The following chart shows the association between a *Job*, *Document Classes* and *Form Types*:

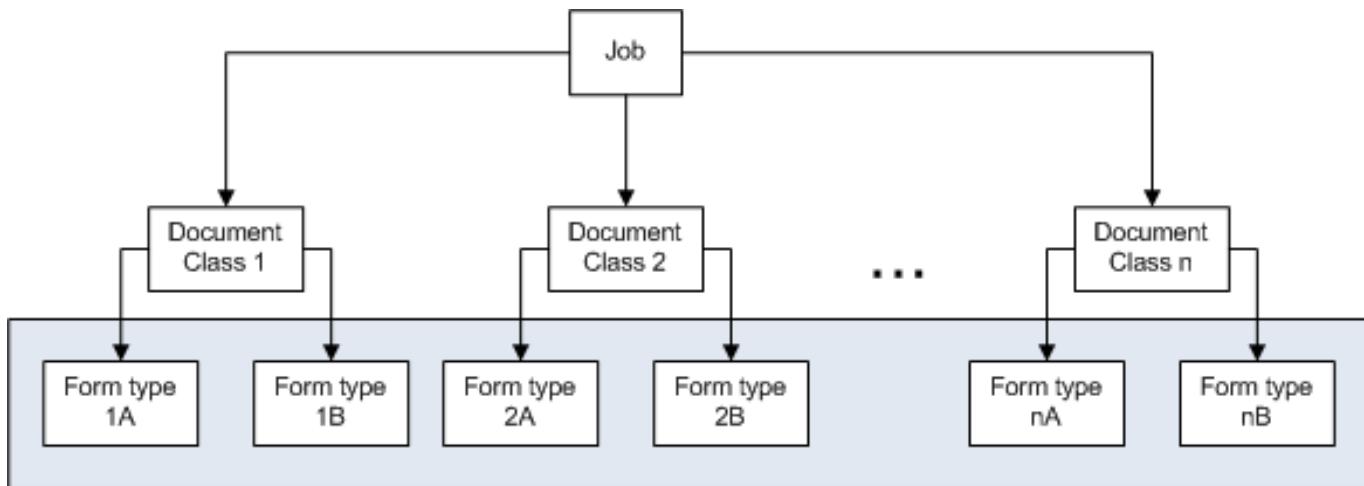


Figure 33. Job, Document Classes and Form Types

3.1.9.1. Batch level indexing

Info Input Solution supports indexing at the *Batch level*, in a similar way it supports indexing for *Folders* and *Documents*. Since there is always one Batch open, indexing on the Batch level does not require the equivalent of a *Document/Folder Class*, since no more than one class is ever required for the current, single Batch. *Index Fields* for the Batch level are defined on the *Indexing & Export* tab of the *Job Properties dialog*, from the *Batch level indexing* button:

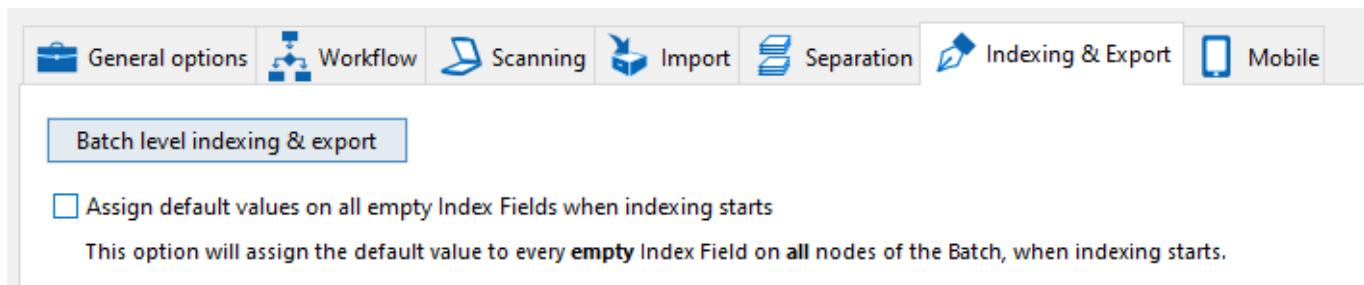


Figure 34. Batch level indexing & export button

Batch level options are identical to the ones from the *Document/Folder Classes* and are discussed in the following chapters.

3.1.10. Mobile

On the *Mobile* tab you can choose which document class(es) should be available to Mobile Client users. Only *Document Classes* already assigned to this *Job* are available here. Note that, the Mobile Client does not use all Batch levels (Batch, Folder, Document, Page), it simply creates document nodes, containing

one or more pages.

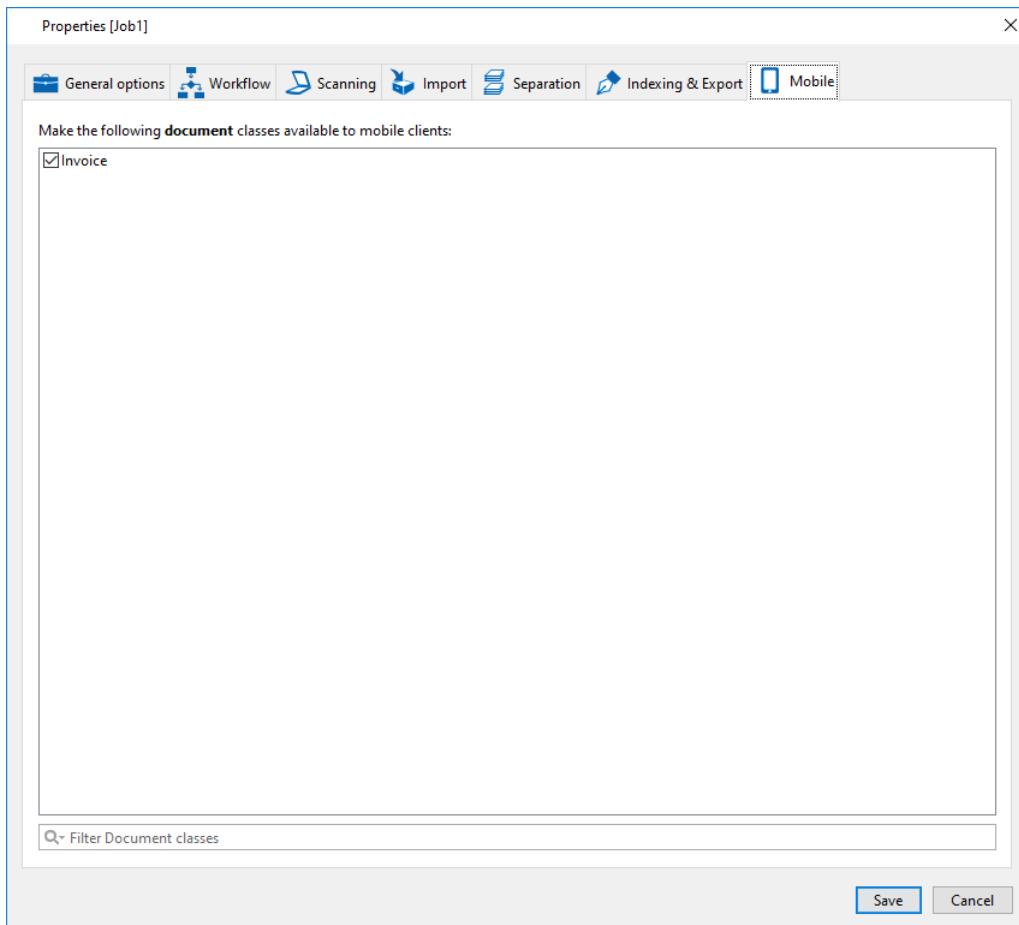


Figure 35. Job Properties dialog: Mobile tab

3.1.11. Workflow Configuration

This section describes the configuration steps necessary to configure the *Workflow* of a *Job* setup.

3.1.11.1. How to edit the Workflow XML

Every *Workflow* consists of workflow *Steps*, modeling the various tasks and processes in the Job workflow. A step can represent a task performed by a human user using the system, an automated task performed by a system agent (service), or an integration point with another system. *Scan*, *Index* and *Export* are examples of this type. There may also be steps that can perform their processing “in-line” with step progression, that is, immediately when the workflow unit completes the previous step and enters the new step. Such steps do not need a human user or a stand-alone subsystem to proceed and their processing is not computationally intensive. Server-side *Branch Steps* are an example of this type.

Each *Workflow* XML must consist of one *Workflow* start-tag and one *Workflow* end-tag, and multiple step

elements, either Linear Type (*LinearStep* XML tag) or Branch Type (*BranchingStep* XML tag).

The following Workflow start-tag attributes can be modified:

- *startStep*: the first step of the Workflow, usually named "Scan". The start step must have the "batch.create" activity attribute. This step name must be defined in the Workflow step elements that follow.
- *description*: this is a free text description of the Workflow design, that will be displayed in the Select Workflow menu.
- *name*: this is also a free text field for the Workflow name.

Every *LinearStep* must contain the following attributes:

- *name*: this must be a unique Step name that should be used by other Steps, when pointing to the next node in the Workflow.
- *displayName*: this is the Step name that will be displayed in the Workflow graphical representation window.
- *next*: this is the Step name of the next Step in the Workflow.
- *security*: this attribute will allow for Step security settings for this Step. The possible values are "USER", for Client-side Workflow Steps that may require additional security settings, and "SYSTEM" for Server-side Steps that are not executed with specific user.
- *activity*: the possible Step activities are
 1. Batch Creation (batch.create activity): Every Workflow must start with a Batch creation activity.
 2. Scan (scan activity): Any additional Scan step, after the first batch creation step, must use this activity.
 3. Indexing (index activity): A Workflow can contain zero, one or multiple Indexing activities. Additional configuration for each Indexing activity is possible.
 4. Server Script (runscript.server activity): A server-side Step script will automatically execute javascript code on the Core Service.
 5. Classify (classify activity): A server-side step that will execute any Classification configuration that is defined in this step. More information about the Classification configuration can be found in the Workflow Step Configuration section below.
 6. Extract (extract activity): A server-side step that will perform Extraction for any index zone that is defined in the document class. Note, this step is mandatory in case the indexing is performed from the HTML Client.
 7. Intelligent OCR (ocr activity): A server-side step that will perform the OCR configuration that is defined in this step. More information about the Intelligent OCR configuration can be found in the Workflow Step Configuration section below.
 8. Fork (fork activity): A server-side step that will split the nodes of a Batch into multiple tasks. According to this step's configuration the Folders or the Documents of the Batch will be split in different tasks. Each task will follow the Workflow separately. See more details in the Workflow

Step Configuration section below.

9. Join (join activity): A server-side step that will join all the previously spitted tasks. All the tasks needs to get in this Workflow step so the batch will continue to the next step. See more details in the Workflow Step Configuration section below.
10. Image Enhancement (image enhance activity): A server-side step that will perform Image operations on the images. See more details in the Workflow Step Configuration section below.
11. Export (release activity): In this step the export destinations will be executed, more than one export steps can be added. See more details in the Workflow Step Configuration section below.



The activity for a [Custom Workflow Actor](#) step must be identical to the activity that has been configured in the *Custom Workflow Actor's Jar file*.

Every *BranchingStep* must contain the same attributes with *LinearStep* above, and must also contain two (or more) *Branch* elements. Every *Branch* element must have the following two attributes:

1. *step*: this is the Step name of the next Step for this branch.
2. *name*: this is the branch name that will be displayed in the Workflow graphical representation window.

Server-side script steps have read-write access to the batch, already uploaded to the Core Service, and the images of this batch. It is possible to use methods for changing the Batch structure, like split or merge Nodes, duplicate Images, detect Barcode data and set Document Form Types.

When the server-side script step is Branch type, the script execution must indicate the branch name that should be followed. Branch Type steps can also be defined for client-side activities (Batch Creation, Scan or Index). An additional script will be available at step level configuration, which will run at client-side, when the user closes the Workflow step.

[Custom Workflow Actor](#) steps can be programmed to perform many different actions. Depending on the Actor Jar file that has been created, and the actions that it performs, certain properties may need to be set in the Workflow XML. This could be handled through code.

Setting Actor properties in the XML

```
<LinearStep next="Export" name="CUSTOM" displayName="CustomActor-SetProperties"
activity="set-properties-activity" security="SYSTEM">
    <Parameter name="batchprop" password="false">batch-value</Parameter>
    <Parameter name="docprop" password="false">doc-value</Parameter>
</LinearStep>
```

Or from the GUI.

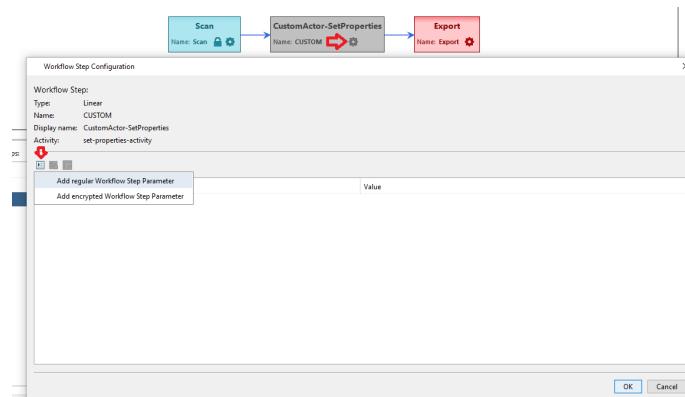


Figure 36. Configuring Actor through the GUI

3.1.11.2. Workflow Steps Configuration

Edit Step

Each Workflow step can be further configured the *Workflow tab* you can select the Job's Workflow from the predefined list of supported Workflow scenarios.

Step Security

Since Job-level security (user groups associated with Jobs) cannot handle the security requirements of certain use cases (e.g. multiple Indexing), you are able to associate user groups with individual steps.

The *Scan* step configuration offers the Force Indexing on the same workstation (before upload). Checking this option will force the operator that performed the Scan to also Index the batch before closing or suspending. If there are no required fields for this batch, Indexing is not forced by design. The *Review* tab is also available, from the *Review* tab it is possible to configure the *Review Functionality*. More information about configuring the Review Functionality can be found in the section [Review/Rescan Functionality](#)

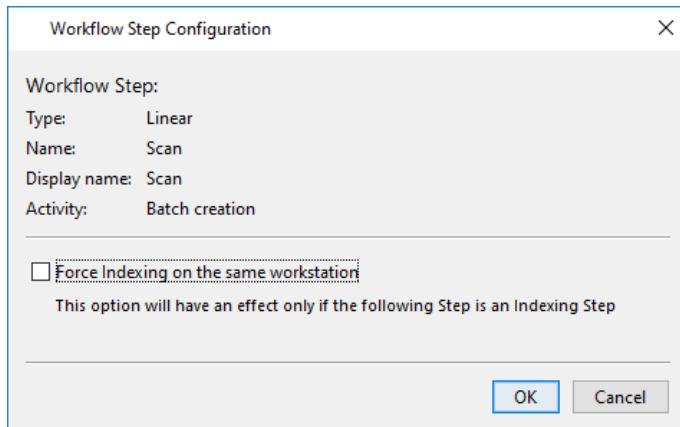


Figure 37. Workflow: Scan step configuration

The *Index* step configuration window, the tabs *Index Classes*, *Advanced Indexing Forms*, *Review*, and *Script* are available,

In the *Index Classes* tab a list all *Job* associated *Index Classes* (at Batch, Folder or Document level) and for each *Index Class*, you can further edit the Index Field properties and the *Load* and *Save* methods.

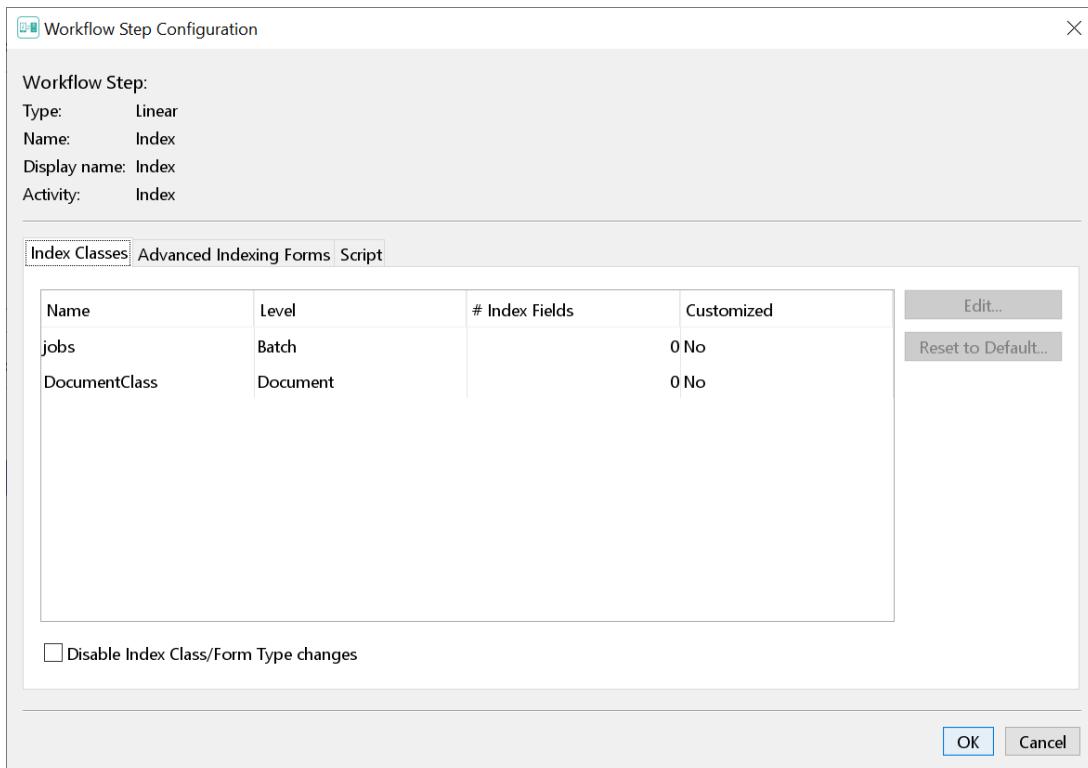


Figure 38. Workflow: Index step configuration

Any changes to the Index Field properties will apply only to the current Indexing Workflow step. A grey box for the *Required* / *Hidden* / *Read-only* properties means that the node's (Batch, Folder or Document) Class level configuration will apply. If you check or uncheck this box, this selection will overwrite the

default configuration, only for this Indexing step.

The *Load method* will be called right before opening this Workflow step in the Client, and the *Save method* will be called right after closing the Workflow step and saving the Batch. There are three options for the Load and Save methods (below).

- default: the default namespace will be used.
- expression: a custom namespace and/or variable can be defined. This functionality is useful for Workflow processes with multiple Indexing Workflow steps.
- script: call the script functions `fieldLoad(field)` and `fieldSave(field)`, defined in the current Indexing step's Index scripting, this script will be applied only in when the batch is opened for index in this specific Workflow step.

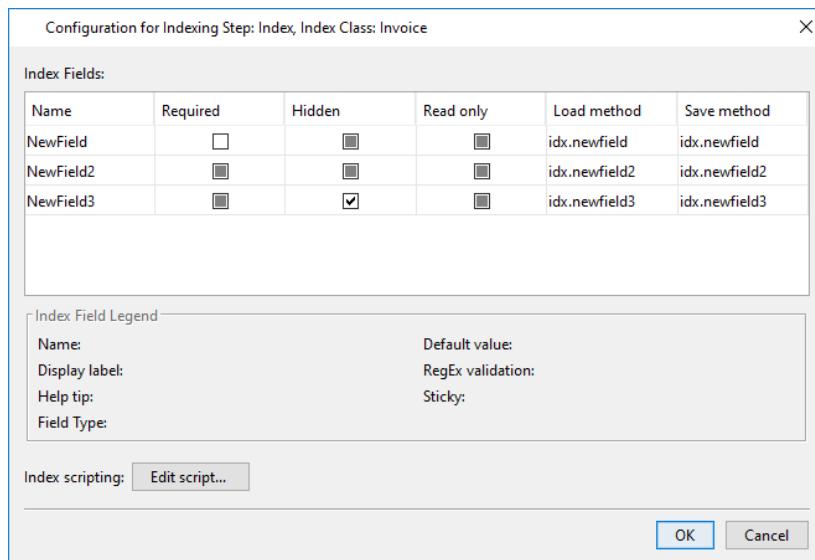


Figure 39. Workflow: Index Class customization for the current Index Step

In the *Advanced Indexing Forms* tab it is possible to define a different custom index forms for this particular index step. Any custom form that is defined in the Workflow will override any configuration that is applied in the document class when the batch is opened in this Workflow step. More information for configuring an advanced indexing form can be found in here [Advanced Indexing Forms](#).

Workflow Step Configuration

Workflow Step:

Type: Linear
Name: Index
Display name: Index
Activity: Index

Index Classes Advanced Indexing Forms Script

Name	Level	Index class advanced form	Step specific advanced form
jobs	Batch	No	No
DocumentClass	Document	No	No
Form Type	Form Type	No	No

Edit Advanced Indexing Form... Delete Advanced Indexing Form

OK Cancel

Figure 40. Workflow: Advanced Index Form customization for the current Index Step

In the *Review* tab it is possible to configure the *Review Functionality*. More information about configuring the Review Functionality can be found in the section [Review/Rescan Functionality](#)

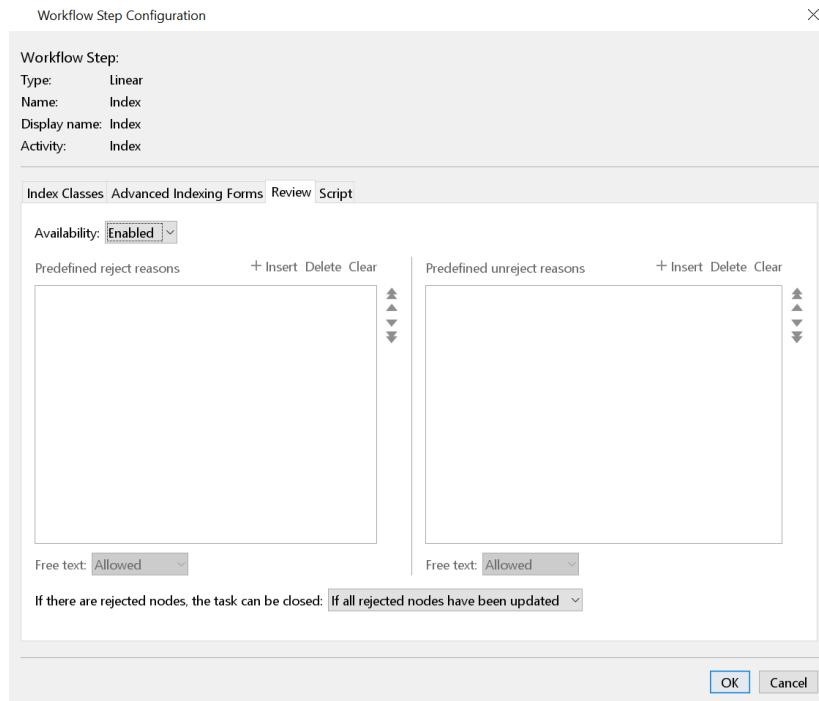


Figure 41. Workflow: Review functionality

In the *Script* tab, it is possible to setup a script that will set the Items in the dropdown list of the Indexing Panel will be available to the user. The node parameter is the **INode** that is currently being indexed and it can be an **IDocument** or an **IFolder**, but not an **IBatch**. If the node being indexed is a *Document* (**IDocument**), then the **allTypes** parameter is an array of **IFormType** objects and if it is a *Folder* (**IFolder**), then **allTypes** parameter is an array of **IFolderClass** objects. The function should return an array of the desired items in the desired order.

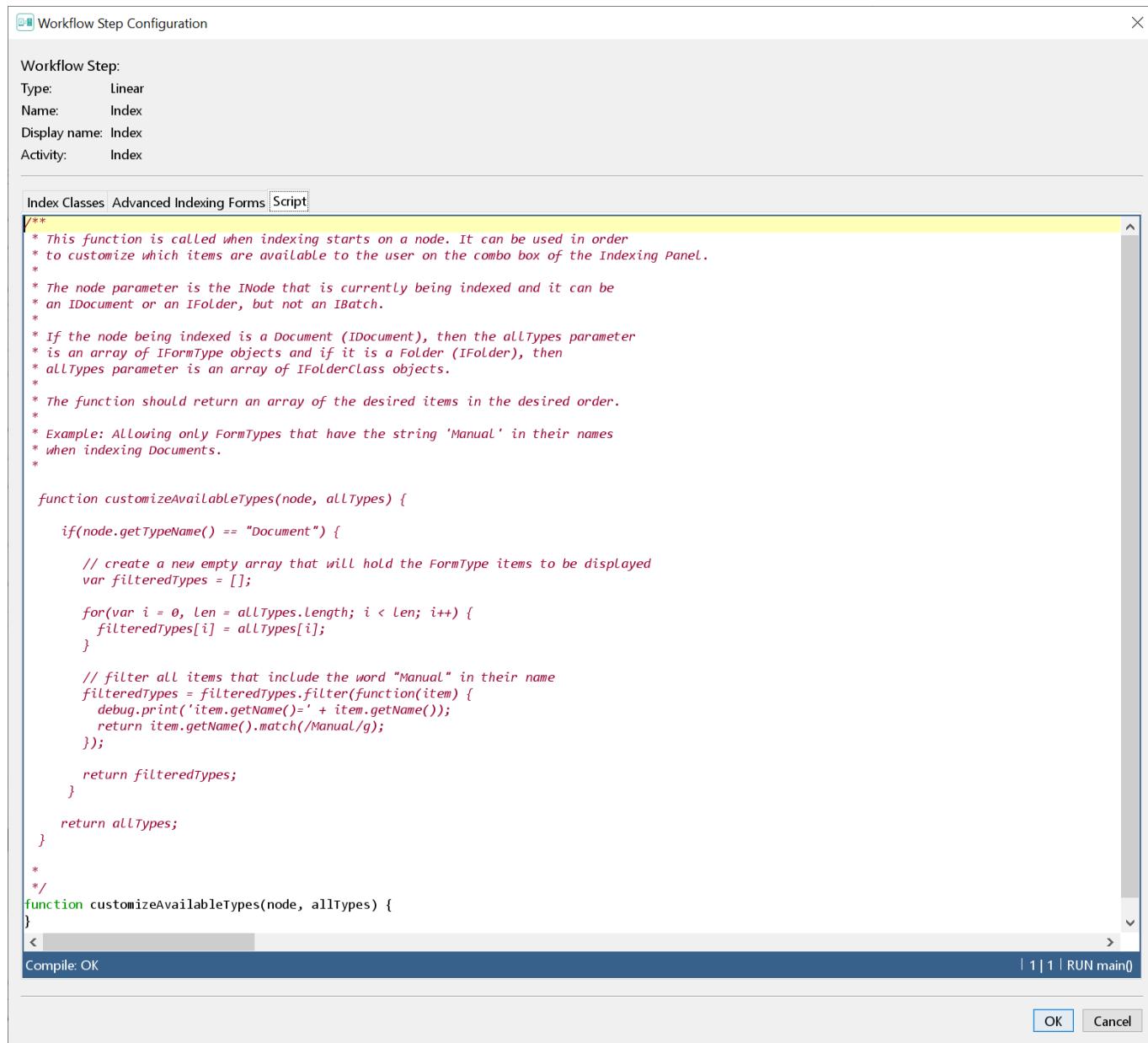


Figure 42. Workflow: Script for the form types, or the folder classes that will be available.

The *Branch* step configuration window contains Javascript code that is executed on the Core Service, automatically, when the workflow process transitions to this step. In *Branch* steps, the function "executeOnServer" must indicate the branch name that should be followed, by returning the name of the next step in the *Workflow* graph.

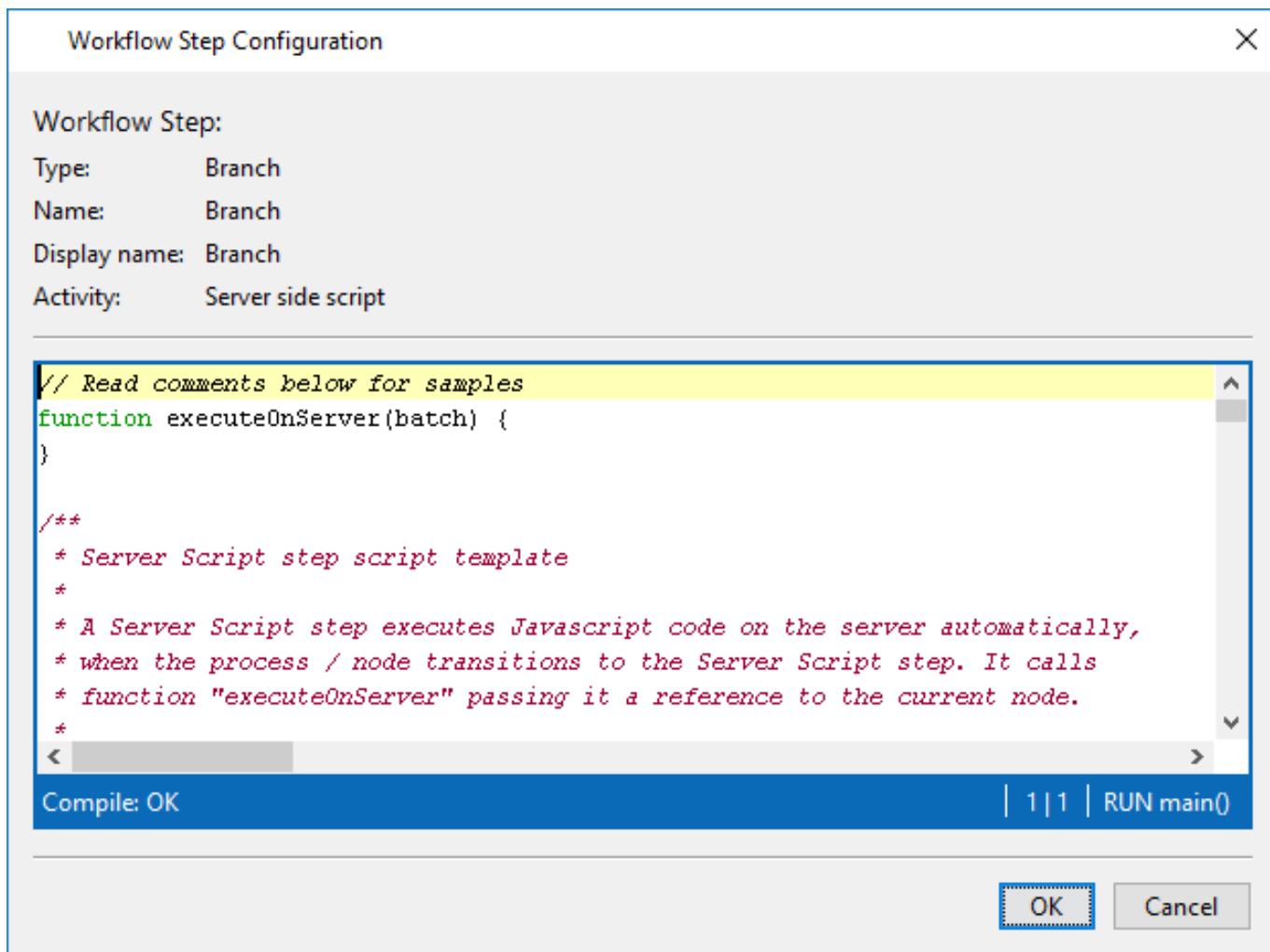


Figure 43. Workflow: Branch step configuration

A workflow does not have to be linear, i.e. a static sequence of steps repeated verbatim every time. There can be parts of a workflow that are executed optionally or mutually exclusively, according to various conditions that are examined at step progression time. In other words, workflows can also be tree-like. Branches enable workflows to be more flexible and suit more scenarios. Branch steps can help in keeping the number of workflows used small, by allowing the workflow designer to design workflows that cover a wide range of scenarios and also expand the coverage of existing workflows. Selecting a branch to follow is a process in itself: conditions have to be examined and a decision made which branch to follow. It is therefore appropriate to use workflow steps for branch-selecting processes.

For Branch steps that are not using the activity Server Script (runscript.server activity) it is required to implement the method `getBranchName()` and return the name of the branch that is to be executed next. The return value may either be a string or an object with a property 'BranchName'. Notice, that the branch name is different than the step name.

The *Classify* step configuration, this step can be used to perform classification, more information about

the classification step can be found in the section [Classification and Extraction Engines](#).

The *Extract* step is used to perform OCR according to the form types configuration, the following options are available,

Run extraction only for Index Fields that are empty

If this checkbox is selected the extraction will run only for the index fields that are empty and assigned in index zones.

Page image selection

This option allows the selection of a specific image for extraction. For example, it is possible to perform image enhancements on an image and then save it with a specific alias, then this alias can be used in this step in order to perform extraction on an image with better quality. This is explained further in the Image Enhancement step configuration below.

On extraction error

This option dictates the behavior that will be followed when an error is occurred. The possible options are the following, Abort: this will abort the operation, Continue: this will ignore the error and continue, Use Script: this will execute the script that is defined in the tab Script in the function `onExtractionError`.

Image registration for extraction

The image registration will use the template image that is defined in the form type and it will try to align the scanned image on top of the template image. This operation improves the extraction accuracy. The available options are the following, Enable: this will perform the image registration, Enable and extract even if image registration fails: this will perform extraction even if image registration fails, e.g. a wrong type of image was scanned. Disable: this will disable the image registration.

Script

The script tab provides access to the functions `fieldExtractionCompleted()` and `onExtractionError()`. The function `fieldExtractionCompleted()` is called whenever extraction is complete for an index field. Note that it is possible to define more than one index zones for the same index field. This function is called once, when the extraction for all related index zones is complete. The function provides a list that contains extracted data from all index zones. As a convenience it also provides the extracted data with the highest confidence value. The default behavior is to assign as the index field value, the extraction result with the maximum confidence. In addition, the review flag for the index field is set, if the maximum confidence is lower than the minimum confidence for the index zone. The confidence of the extraction result is saved as a property of the document, with name `sys.extraction.<index_field_name>.confidence`. When this function is called, the default behavior has already been applied and the index field has been filled. You may customize the behavior of the extraction by using the appropriate method on the document or index field. The function `onExtractionCompleted()` is called whenever extraction is complete for an index field. Note that it is possible to define more than one index zones for the same index field. This function is called once, when the extraction for all related index zones is complete. The function provides a list that contains extracted data from all index zones. As a convenience it also provides the extracted data with the highest confidence value. The default behavior is to assign as the index field value, the extraction result with the maximum confidence. In addition, the review flag for the index field is set, if the maximum confidence is lower than the minimum confidence for the index zone. The confidence of the extraction result is saved as a property of the document, with name `sys.extraction.<index_field_name>.confidence`. When this function is called, the default behavior has already been applied and the index field has been filled. You may customize the behavior of the extraction by using the appropriate method on the document or index field. The function `onExtractionError()` is called whenever an error occurs during extraction. The function provides the error message and the index field that failed. You may customize the behavior of the extraction by using the appropriate method on the document or index field.

`tionError()` is called whenever extraction fails for an index field and the continue on error checkbox is unchecked. The function is called for each index zone defined for the index field. You may abort the rest of the extraction process by returning the value `Const.AbortOp`; This function can be used in order to create more complex extraction abortion logic, for example aborting only after X errors or if more than 50% of the extraction operations failed.

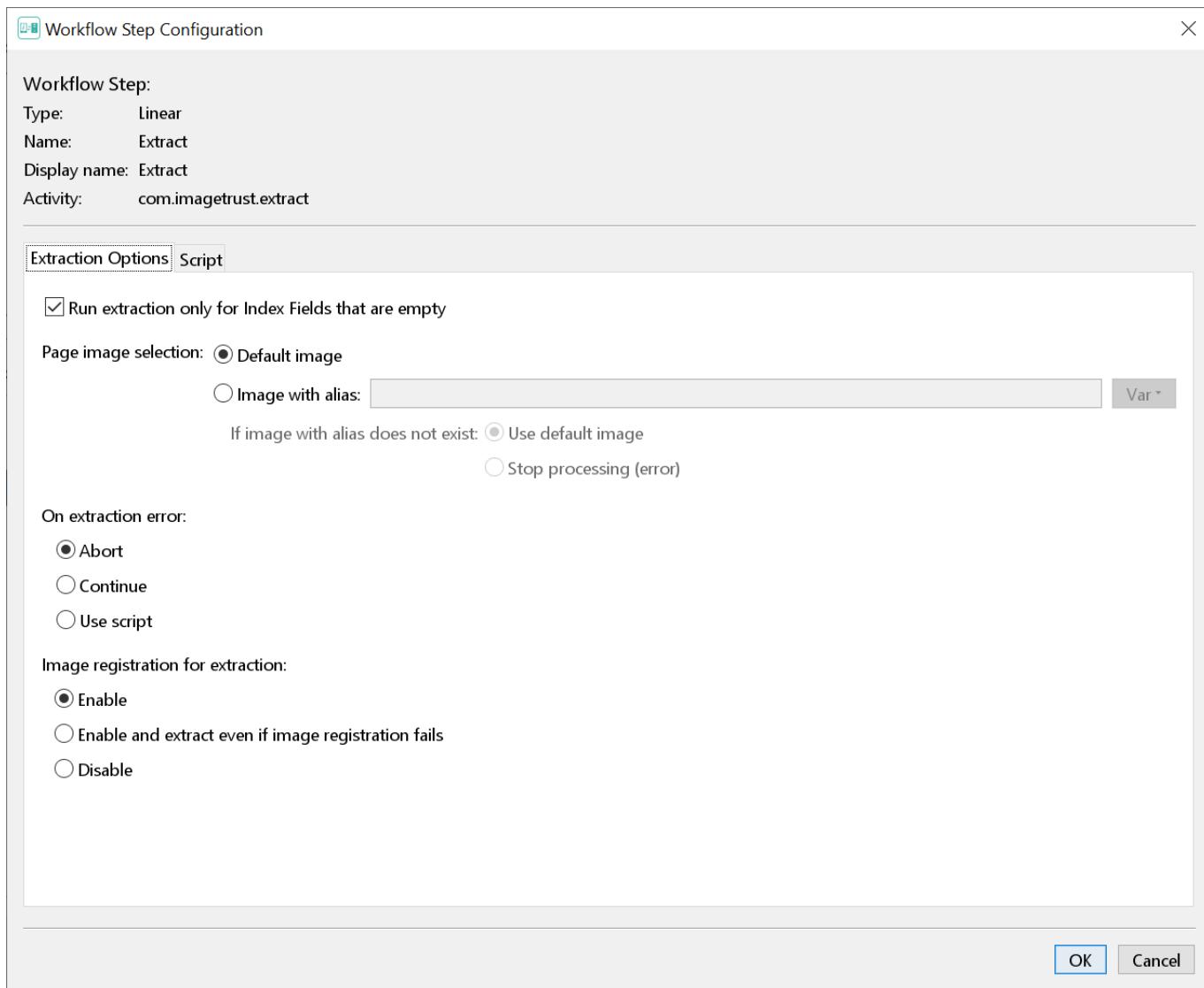


Figure 44. Workflow: Fork Join Workflow example

The *Intelligent OCR* step is an automated document recognition and extraction module that uses Machine Learning technology in order to identify specific document types and structures, such as invoices and forms, or perform full text OCR in documents. More information about configuring this step can be found in the section [Intelligent OCR Engines](#).

The *Fork* step can be used to split the nodes of a *Batch* into multiple *Tasks*. According to this step's configuration the *Folders* or the *Documents* of the *Batch* will be split in different *Tasks*. Each *Task* will follow

the Workflow separately. The available options are Document: this will split the *Documents* of the *Batch* into separate *Tasks*. Folder: this will split the *Folders* of the *Batch* into separate *Tasks*. After this step each task will go through the Workflow steps separately. For example, the index step will provide access only to the specific *Tasks* and not on all the *Nodes* of the *Batch*. For steps that use the activity Server Script (runscript.server activity) a *Batch* element will be provided in the parameter but only access to the specific task will be allowed.

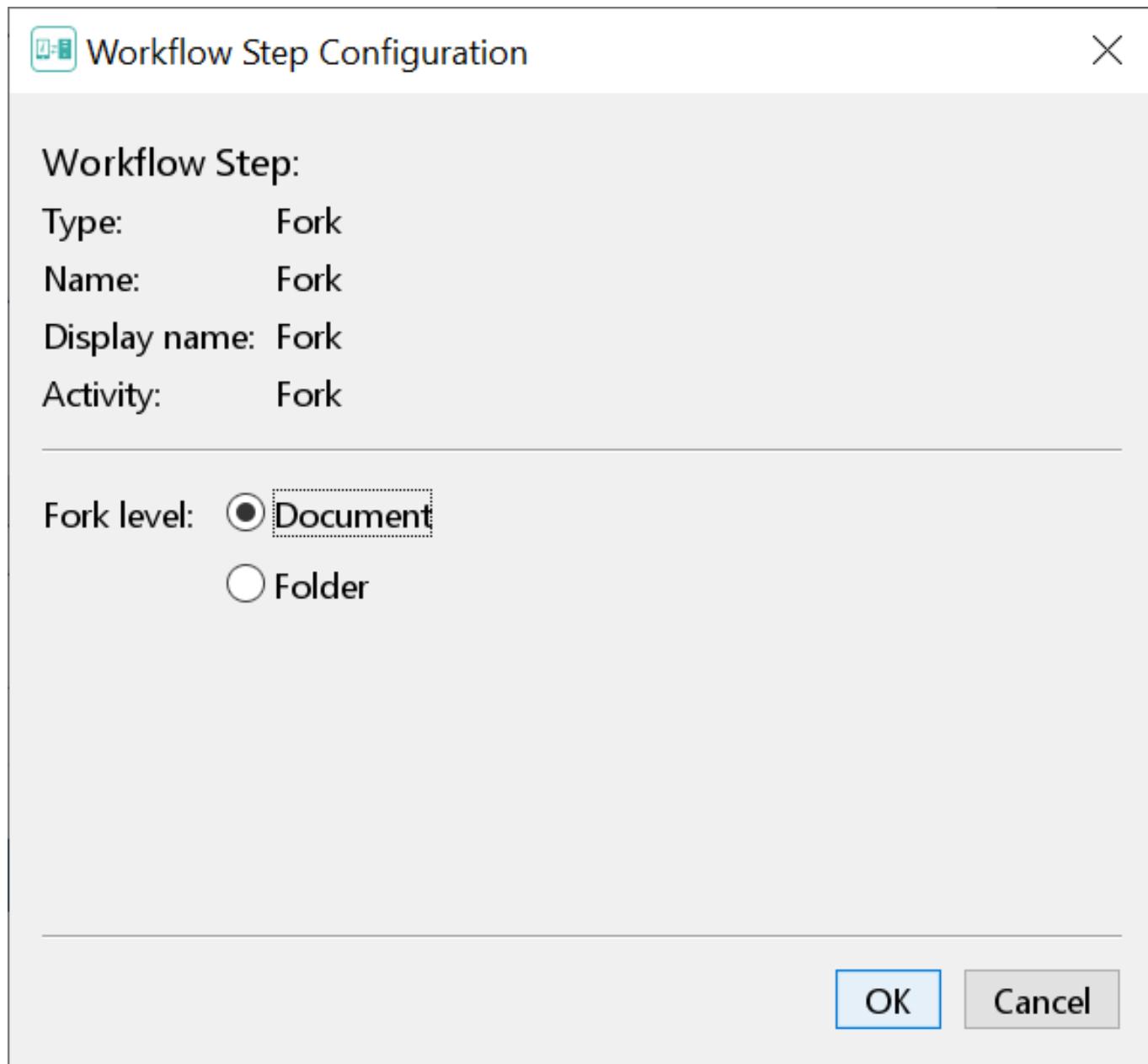


Figure 45. Workflow: Fork Step configuration

The *Join* step can be used only after a Fork step. This will merge the *Tasks* of the *Batch* back together after all the tasks have followed the Workflow and arrived to this step. For example, in the Workflow

below the batch will be merged again only after all the tasks have been exported successfully. No configuration is available for this step.

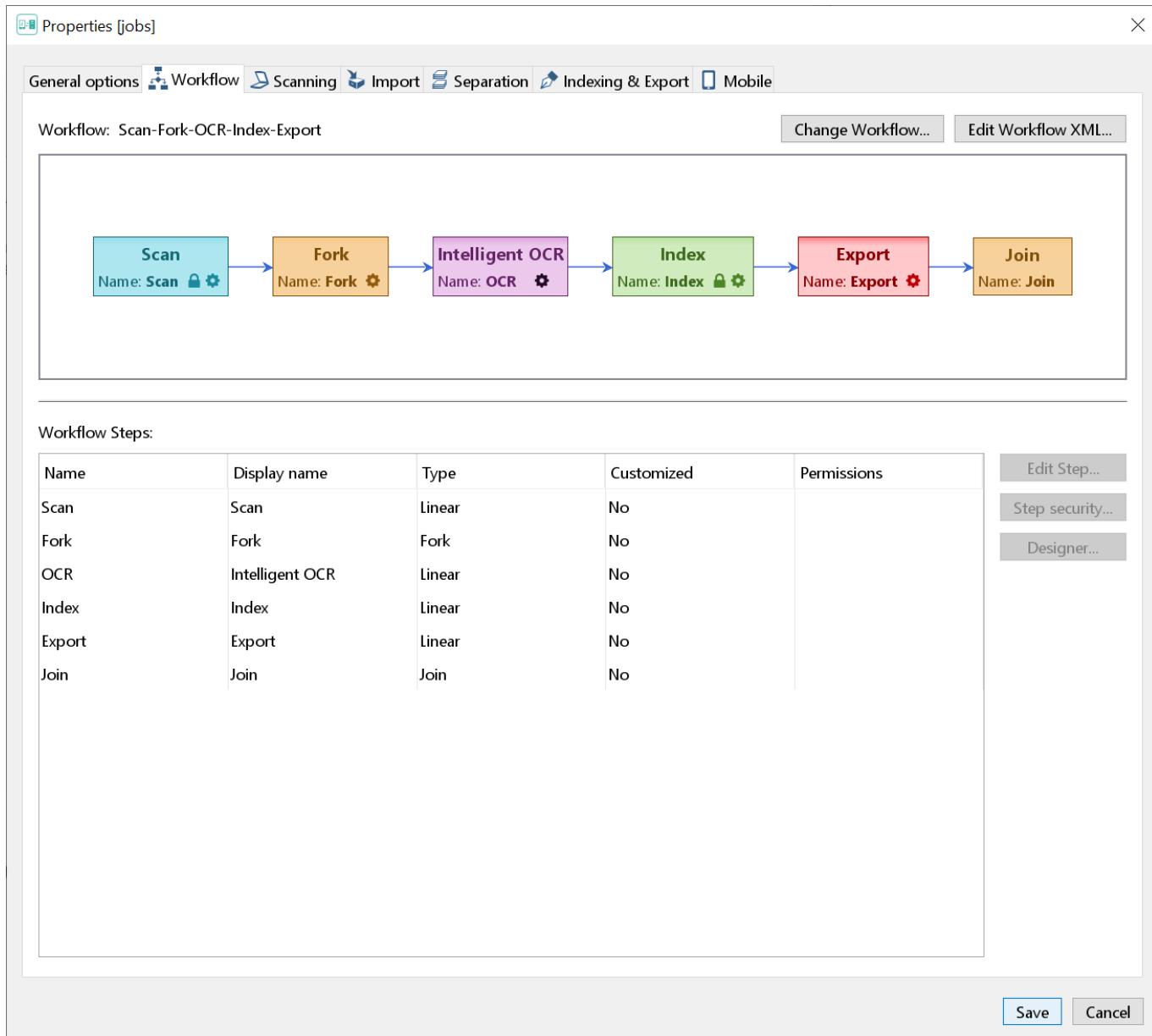


Figure 46. Workflow: Fork Join Workflow example

The *Image Enhancement* step is a server-side step that will perform Image operations on the images. The available options are, *Image Enhancement Profile*: this is the Image Enhancement profile that will be used in the target pages. Select the button *Edit Image Enhancement Profiles...* to create a new profile. More information about the Enhancement profile can be found in the section [Image Enhancements profiles](#). *Page Range*: All, this will perform the image operations in all pages. *Pages*: the index of the pages that will be used in the image operations, for example, if set to 1 that means that the first page of each document will be used.

Workflow Step Configuration

Workflow Step:

Type: Linear
Name: ImageEnhancement
Display name: Image Enhancement
Activity: com.imagetrust.imageenhance

Options

Image Enhancement Profile:

Page range:

All
 Pages:

*Enter a comma separated list of pages or page ranges
For example: 1,2,5-8*

OK Cancel

The *Export* step configuration window list all *Job* associated *Index Classes* (at Batch, Folder or Document level).

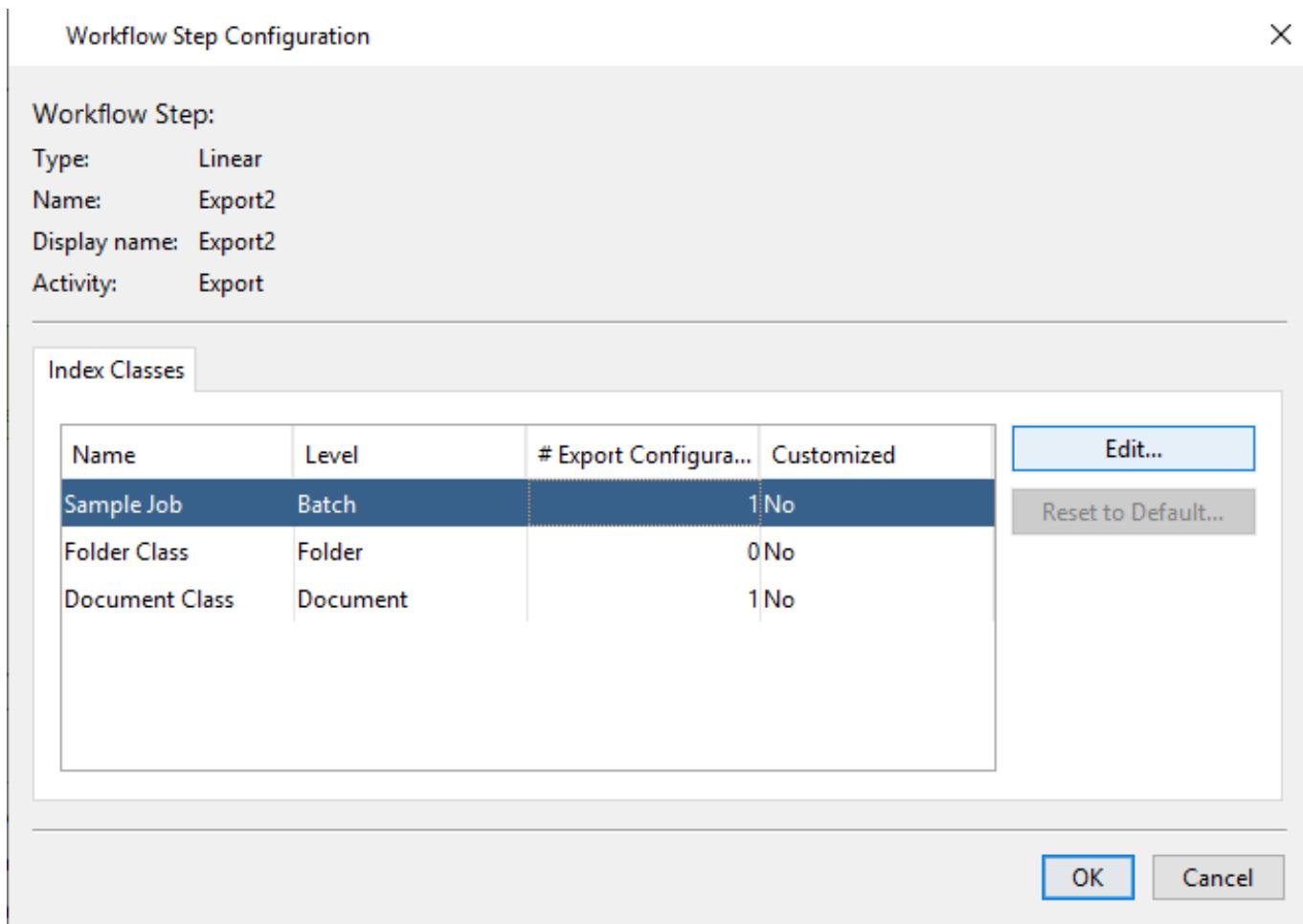


Figure 47. Workflow: Export Step configuration

Inside each Index Class the current export destinations configuration is shown. From this window more export destinations can be added for the current Export step by using the Add button. The Already configured export destinations are shown in bold and they cannot be changed. Only export destinations that are added from the workflow can be changed from this window. The column Active is used to disable\enable an export destination that have been set from the Index Class. The checkbox in the column active has three states

- Checked: If the checkbox is checked then the export destination will be used in this Export step
- Unchecked: If the checkbox is unchecked then the export destination will not be used in this Export step
- Default: The default behaviour will be followed. If the export destination is active in the *Index Class* then it will be executed otherwise it wont.

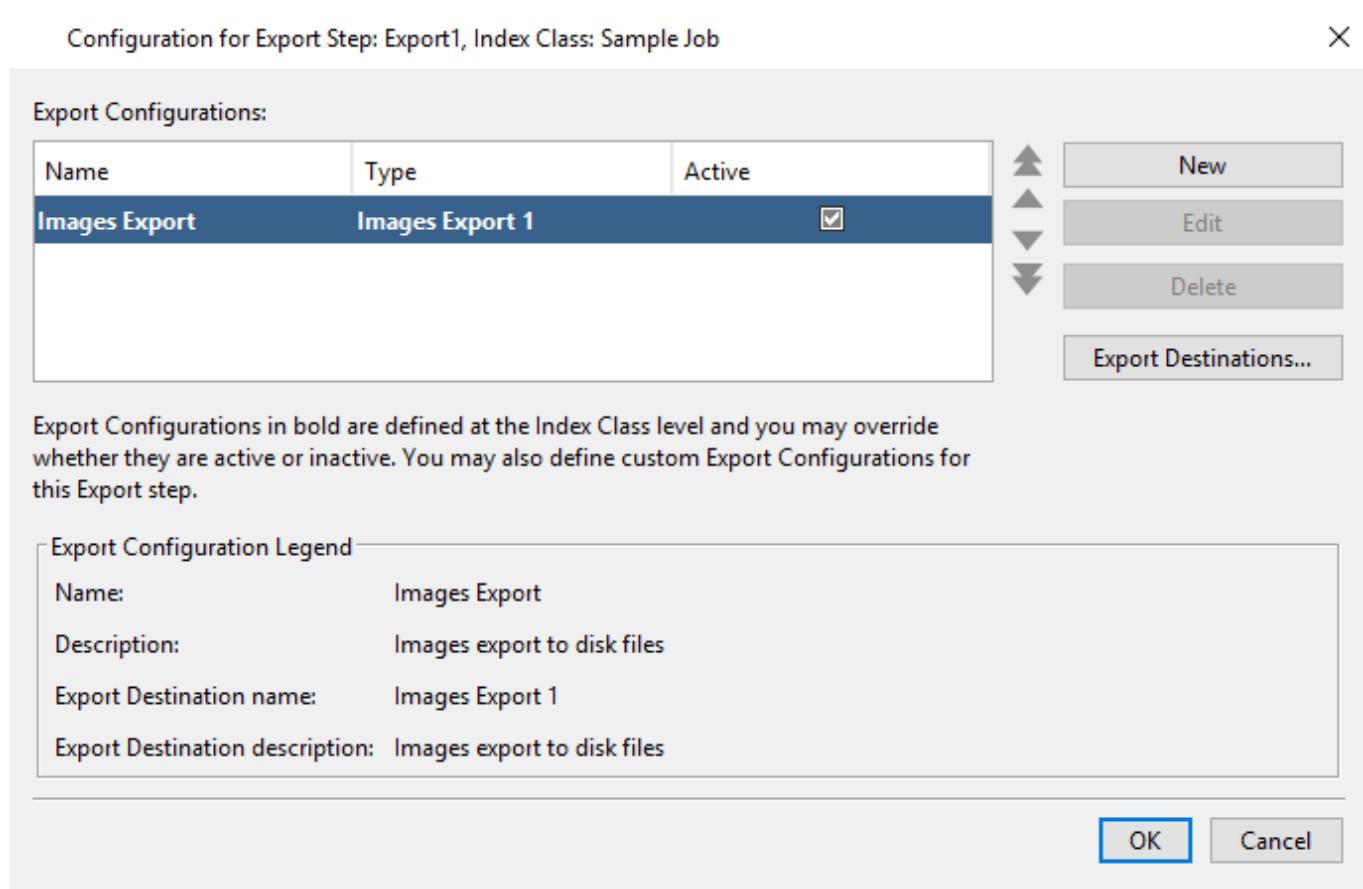


Figure 48. Workflow: Export Step export destination configuration

3.1.11.3. Multiple Indexing Steps

With multiple Indexing steps, in essence we divide the Indexing work into more than one steps, but the system objective remains unaltered: the batch / folder / document must still be indexed. Various Multiple Indexing scenarios can be conceived, where each operator-driven Indexing step contributes only a part of the Indexing work, but none of them the total Indexing work. To support such scenarios, it will be necessary to have non-operator-driven Indexing steps that would automatically collect the pieces of Indexing work performed in the various operator-driven steps and commit them as the final Indexing work, at the same time letting the batch / folder / document progress in the workflow.

Obviously, these steps should also perform the necessary checks to determine whether the Indexing work has been completed and route the batch / folder / document accordingly. Therefore, these steps should be implemented as Branching steps.

The following Use Case examples are supported:

Index and Verify (2 Index steps)

First user does the indexing of the document, second person does verification (only x% of the docu-

ments go to this step).

3-way Indexing (3 Index steps)

Two users will be indexing the same document (both users see the same document with empty fields). If the fields don't match then we route it to a third queue.

Also the following indexing configuration requirements are supported:

Per-step Index Field Values

All Index field values provided in all Indexing steps must be stored, so that they are accessible at any time / step in the workflow. A second Indexing step in the Workflow may need to compare its own value for a field with the value of a previous Indexing step. A decision-making step (e.g. a step that determines whether another Indexing step should follow) must have access to all values of all Indexing steps.

Step-level Security

It must be possible to configure access of user groups to individual steps in a workflow. It must be possible to support use cases where a low-privileged indexer who is permitted to access a low-security Indexing step, is not permitted to access a high-security Indexing step. Our current workflow-wide security model does not support this.

Step-level Indexing Configuration

It is possible for each Indexing step to hide some Index fields, do not show previously entered values, mark fields as read-only for this particular step.

Allow / deny exiting Indexing mode

Exiting Indexing mode (and getting into Scanning mode) effectively allows batch structural changes. This supports use cases where Indexing operators should not be permitted to do that (e.g. move pages / documents around, delete pages, scan new pages, etc.)

Indexing Conclusion and Check Steps

It is possible to include steps in the workflow that would aggregate the results of the Indexing steps preceding them and make decisions on the completeness / correctness of the Indexing activity. They could then route the batch / folder / document to the proper workflow branch, e.g. include an additional Indexing step or proceed to the workflow step.

3.1.11.4. Classification and Extraction Engines

A server-side Classification and Extraction Job Workflow is an automated server-side document recognition project. This project uses multiple workflow processing steps for (a) image classification, (b) document identification and (c) data extraction. Each one of these processing is described in details below.

Setting up a project

Setting up an automatic *Classification and Extraction* project requires several configuration steps. Although not mandatory it is strongly recommended to follow the configuration order outlined below. There are multiple intertwined components that will have to be set up and this order not only minimizes the back and forth between configurations but also serves as a guide for proper preparation when setting up an automatic recognition project.

Step 1: Understand handled content

The first step should be understanding what content will be handled by the Job and gathering the requirements for the project. This includes:

Getting a list of all the types of documents that will be handled by the Job;

For each type of document, get a sample image for as many pages of the document as possible;

Identify which parts of the sample image correspond to fields that make up the document index data.

Step 2: Gather the required samples

Classification projects require an ample amount of sample images for each of the document pages that will be handled by the system. It is generally advised to gather as many samples as possible and split them in two groups, the Training Set and the Test Set. A good rule of thumb is the 80/20 principle, i.e. 20% of all the samples go to the Training Set while the rest to the Test Set. Different versions of the image include (where it applies):

- Unfilled forms
- Filled
- Partially filled
- Off-center shifted
- Somewhat skewed
- Badly scanned images

and in general any other potential scenario that is prevalent in the content that will be handled by the system. The main direction for the type of samples that should be used is always given by the content at hand. If the use case enforces near-perfect scanning of all incoming paper then skewed or off-center samples might not apply for example.

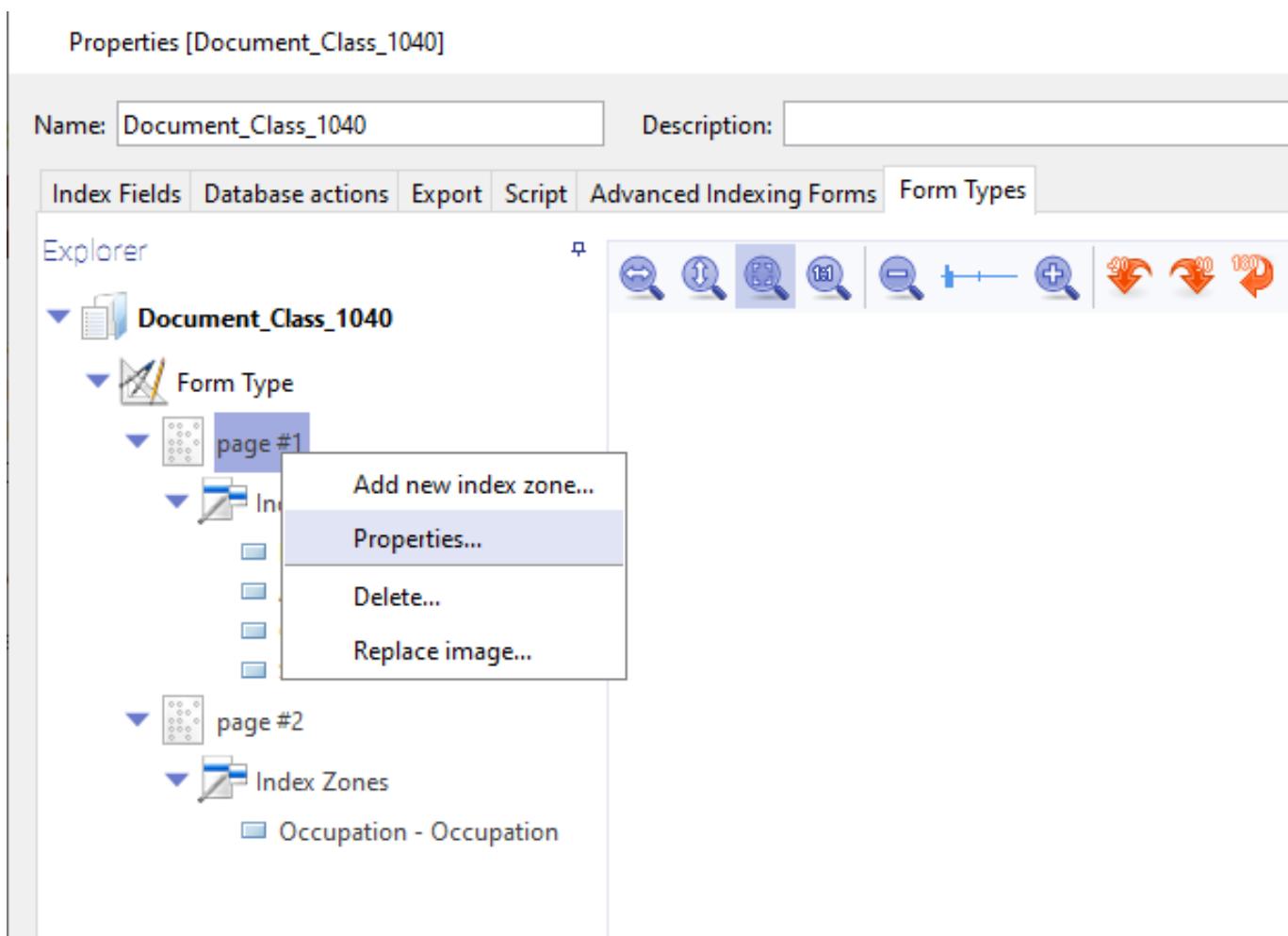
Step 3: Recreate the content structure in Info Input Solution

Once all the requirements and sample images have been collected, they must be represented as Document classes.

1. Create a Document class for each document type;
2. For each field, identify the type of data it contains and create the appropriate Field Types (e.g. String, Date, Yes/No);
3. For each field, define the respective Index Field;
4. Set up the Document class Form Type and add all the Sample Pages available for the specific document;
5. Add the index zones for each of the defined index fields.

Regarding Document class and Index Field setup, refer to the following Sections.

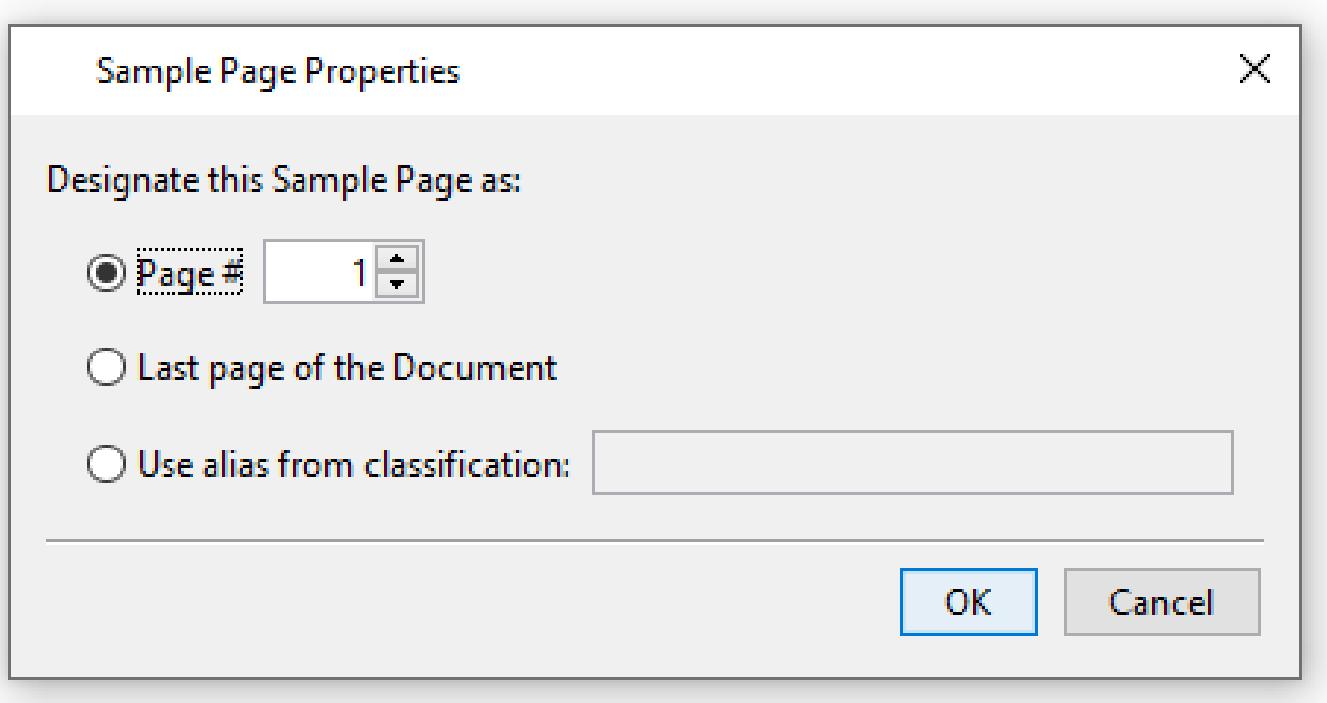
When a Form Type is ready, for each defined sample page its properties must be configured. The properties dialog is available via the context menu which appears by right-clicking on a sample page node in the Form Types tab in the Document class properties dialog.



The sample page properties dialog has 3 options:

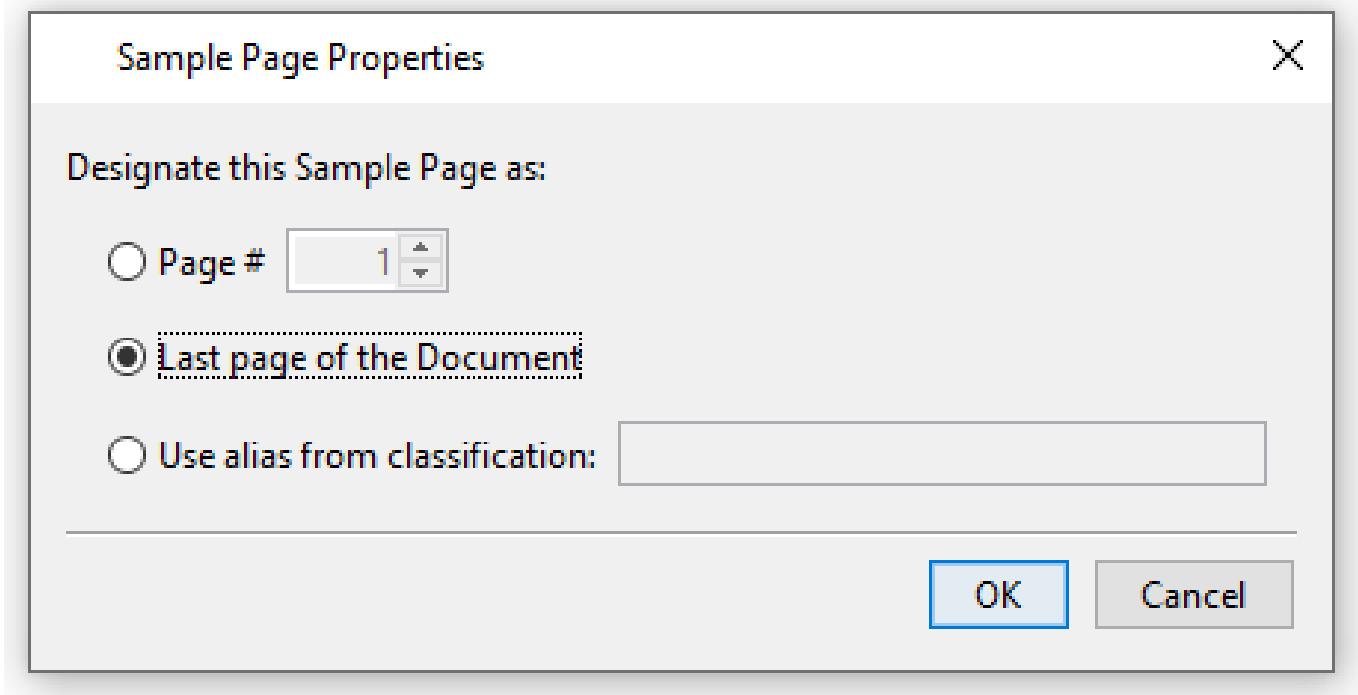
1. Declaring the page's absolute position in the document.

Use this if the absolute position of the page in the document is known beforehand. Knowing this, will allow the proper identification of the Form Type of the Document if all pages are in their respective positions. The first page of a Document is a special case. During the Document Separation step, if a first page is identified, separation will occur at that point. Note that Document Separation will occur only during the Classification workflow step since identifying a loose page as a specific page of the Form Type requires image classification.



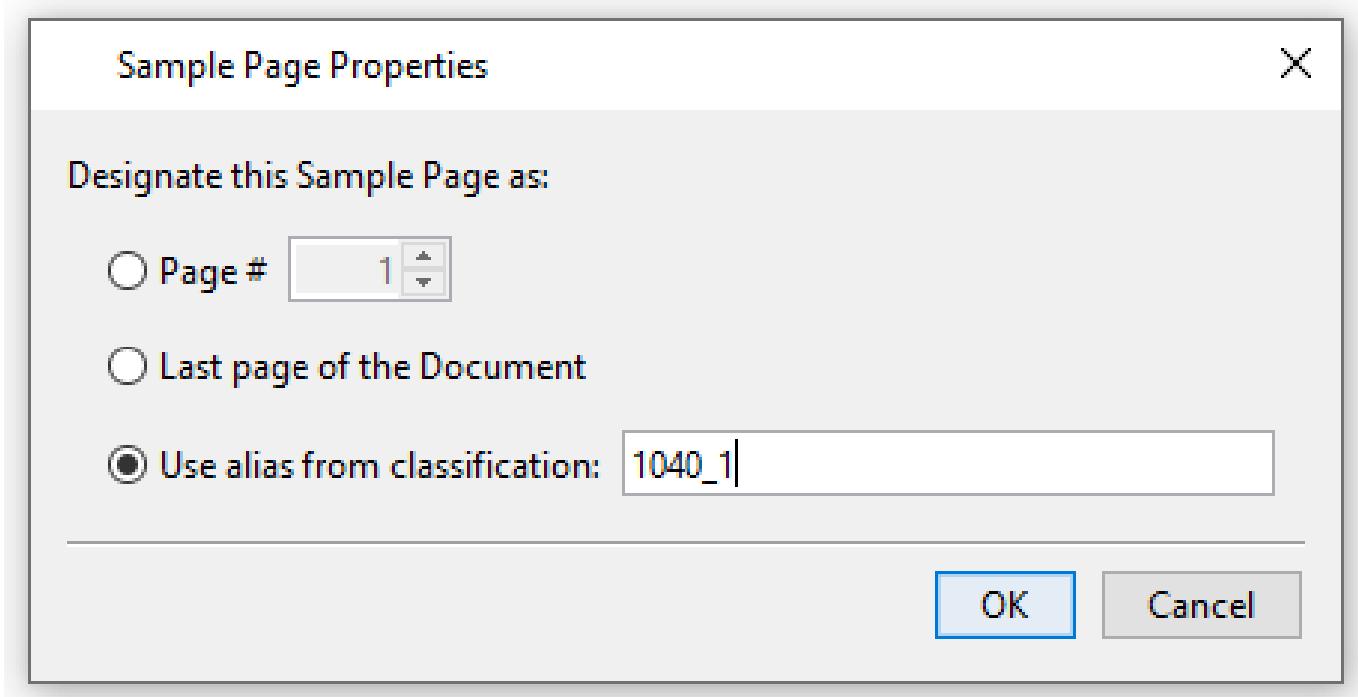
1. Declaring the page as the last page of the document.

It is possible that a document might not have a predefined amount of pages. If however its last page is always the same, then this option allows for declaring it as so. Again this is a special case and separation occurs when such a page is identified. Similarly as above, Document Separation requires a Classification workflow step.



1. Binding the page with a classification alias.

If extraction zones are defined for middle pages (i.e. not the first or last page) then these pages must be given a classification alias. This string essentially pairs the sample page with a respective image type that will be defined during the configuration of the Classification workflow step. View the paragraph Understanding the relationship between the Form Type pages and Classification step images after going through the Classification workflow step chapter.



This option is not supported for the rest API intelligent services.

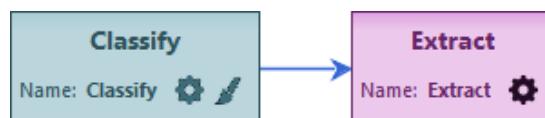
For the rest of this guide we assume the following configuration scenario:

- 3 Document classes, namely 1040, 2106, 2441, each with one Form Type defined
- 1040 has 2 sample pages with absolute positions 1st and 2nd
- 2106 has 2 sample pages with classification aliases 2106_1 and 2106_2
- 2441 has 1 sample page with absolute position Last

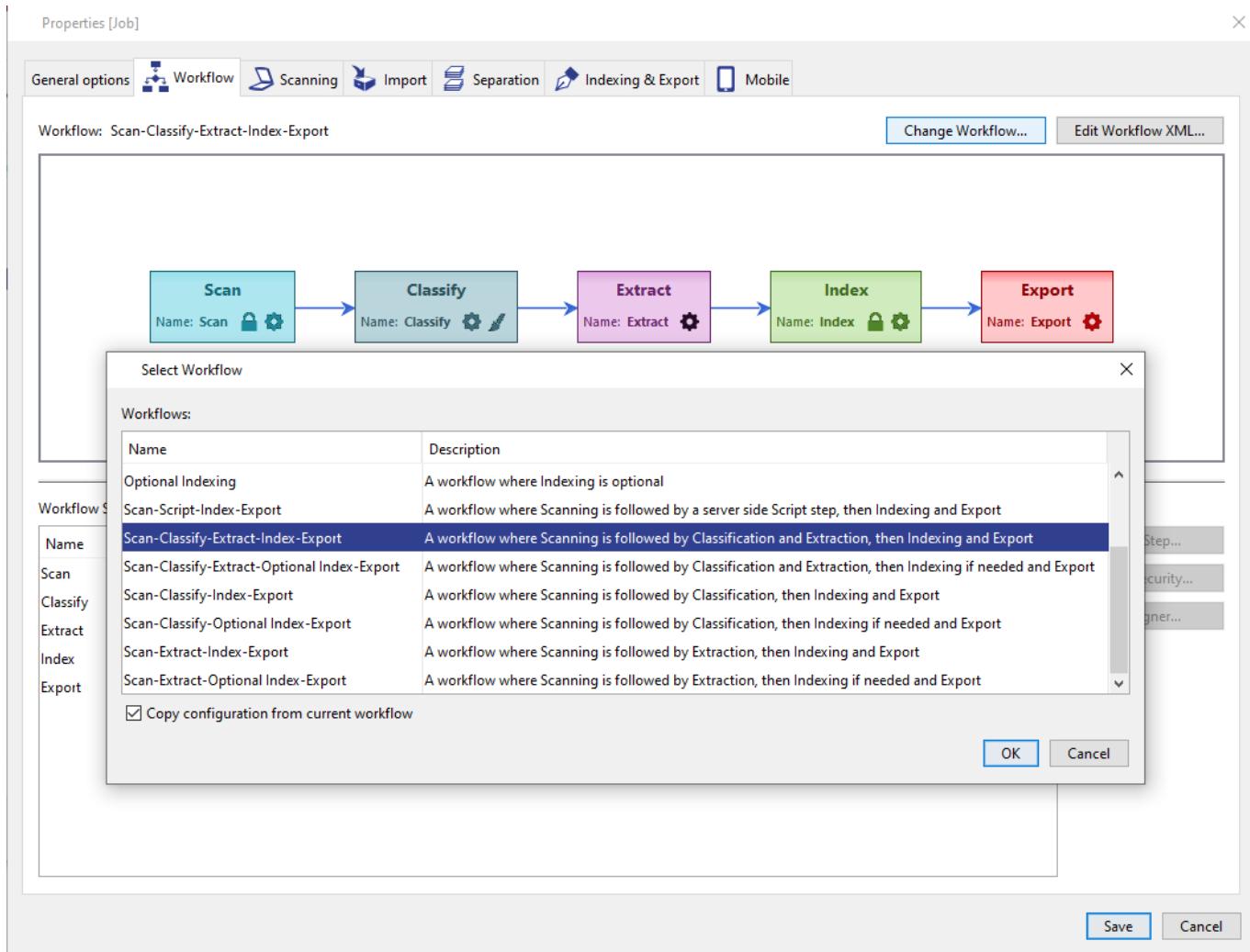
Each Document class has some index fields defined such as Name or SSN.

Step 4: Setup a Classify-Extract workflow

The classification and extraction workflow steps are available during job configuration. Although not mandatory, classification would usually come before extraction. This is because during classification, the Document Form Type will be identified and assigned to it. This is what allows the assignment of the proper index fields to the Document and thus correct extraction. If no Form Type is assigned to a Document when it reaches the extraction step, no extraction is attempted and the Document proceeds further, awaiting possible manual assignment of its Form Type during the Index step.



The quickest way to setup a Classification/Extraction workflow in the Job Workflow tab is by selecting one of the preset workflows by pressing the Change Workflow... button.



The steps are set up and ready to be configured.

Step 5: Classify and Extract step configuration

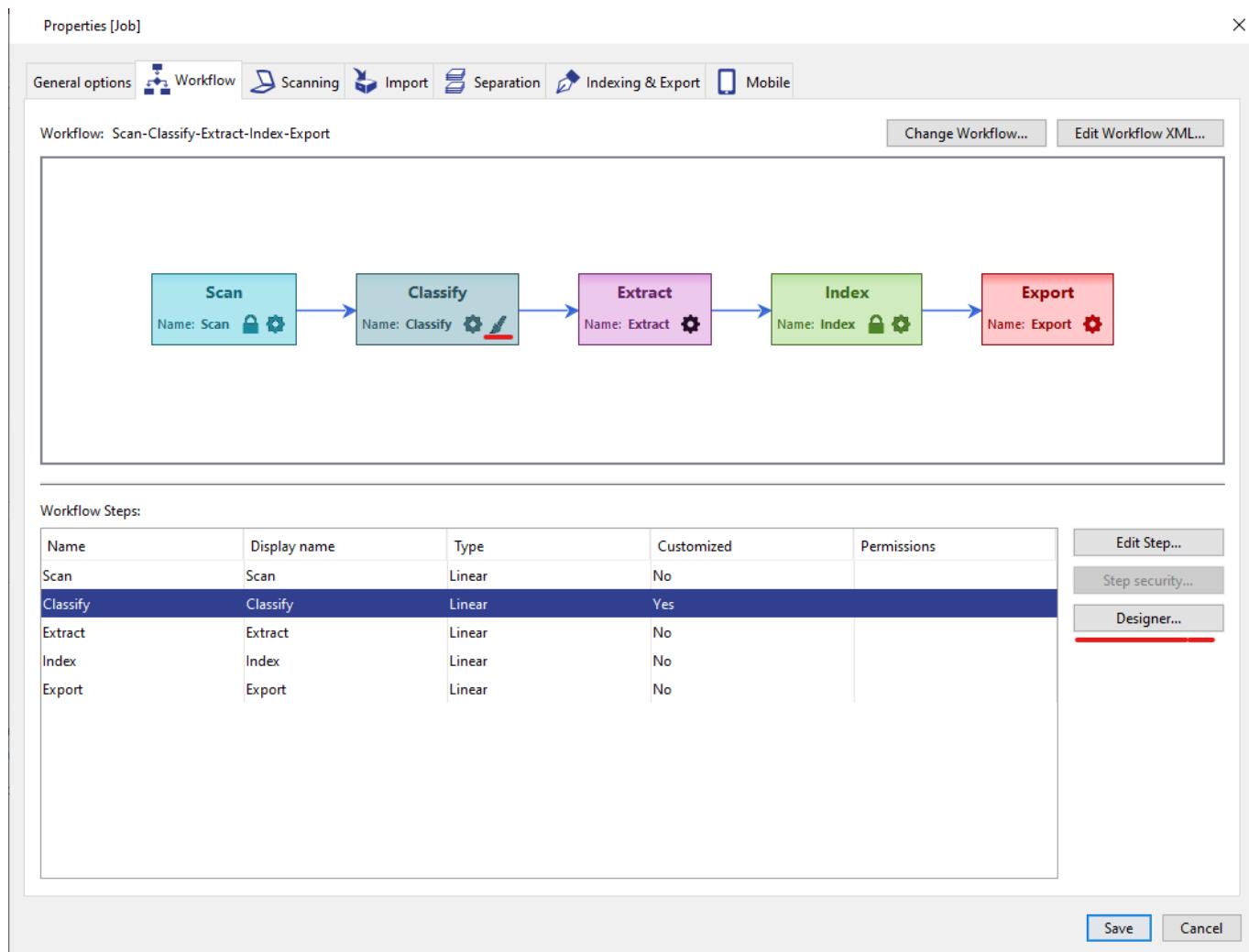
See the section below how to configure the Classify and Extract step.

3.1.11.5. Creating a Classification Project

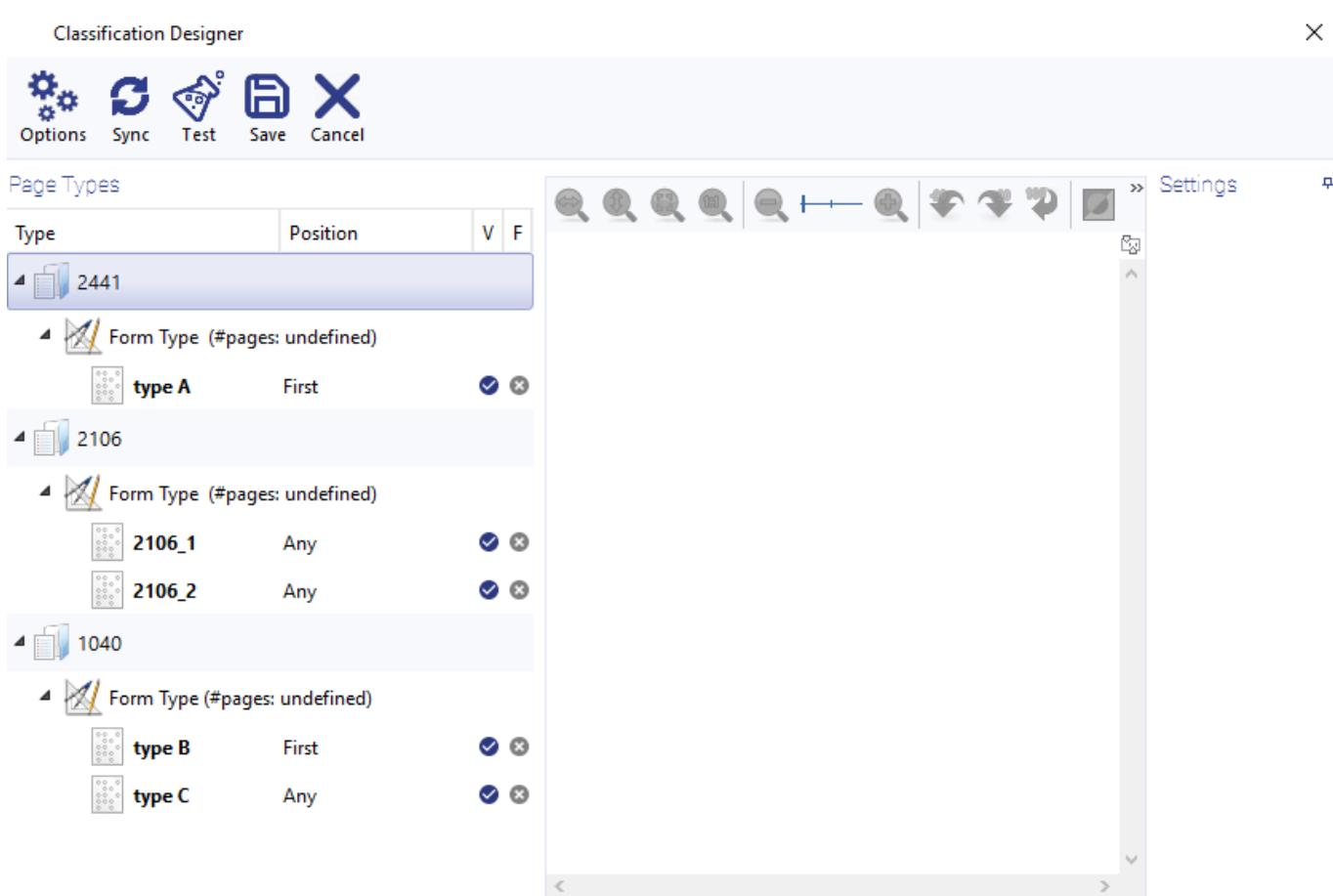
The configuration of the Classify step is the most important part of a successful automatic recognition workflow since a high classification rate means lower manual indexing. In order to achieve this, it is paramount to understand the intricacies of how classification works and how this applies to each document page that will have to be classified by the system. Furthermore, the configuration must be approached holistically, combined with trial and error in order to achieve the highest optimization pos-

sible for the specific use case.

In order to open the Classification Designer, either click on the Classify step brush icon or on the Designer... button with the Classify step selected in the Workflow Steps grid.



Assuming the Document class configuration outlined earlier, the Designer dialog opens, populated with 3 Form Types and their respective Page Types.



When the *Classification Designer* is opened for the first time a *Classification Project* is created and it is Synced, generating the appropriate types for all of the Job's Document classes (Form Types and Sample Pages). Once the Designer has synced with the Document classes, no other link is maintained after that point. It is possible to totally mutate the classification types or the Form Types and no change will be reflected either way. This is very important because one might be lead to believe that changing a Sample Page will be immediately reflected in the Classification Project which is not the case. Maintenance of both sides is required. This is why the Sync operation is also available via the Sync button. This is handy when a Document class changes drastically and it has to be re-imported in the Designer for configuration.

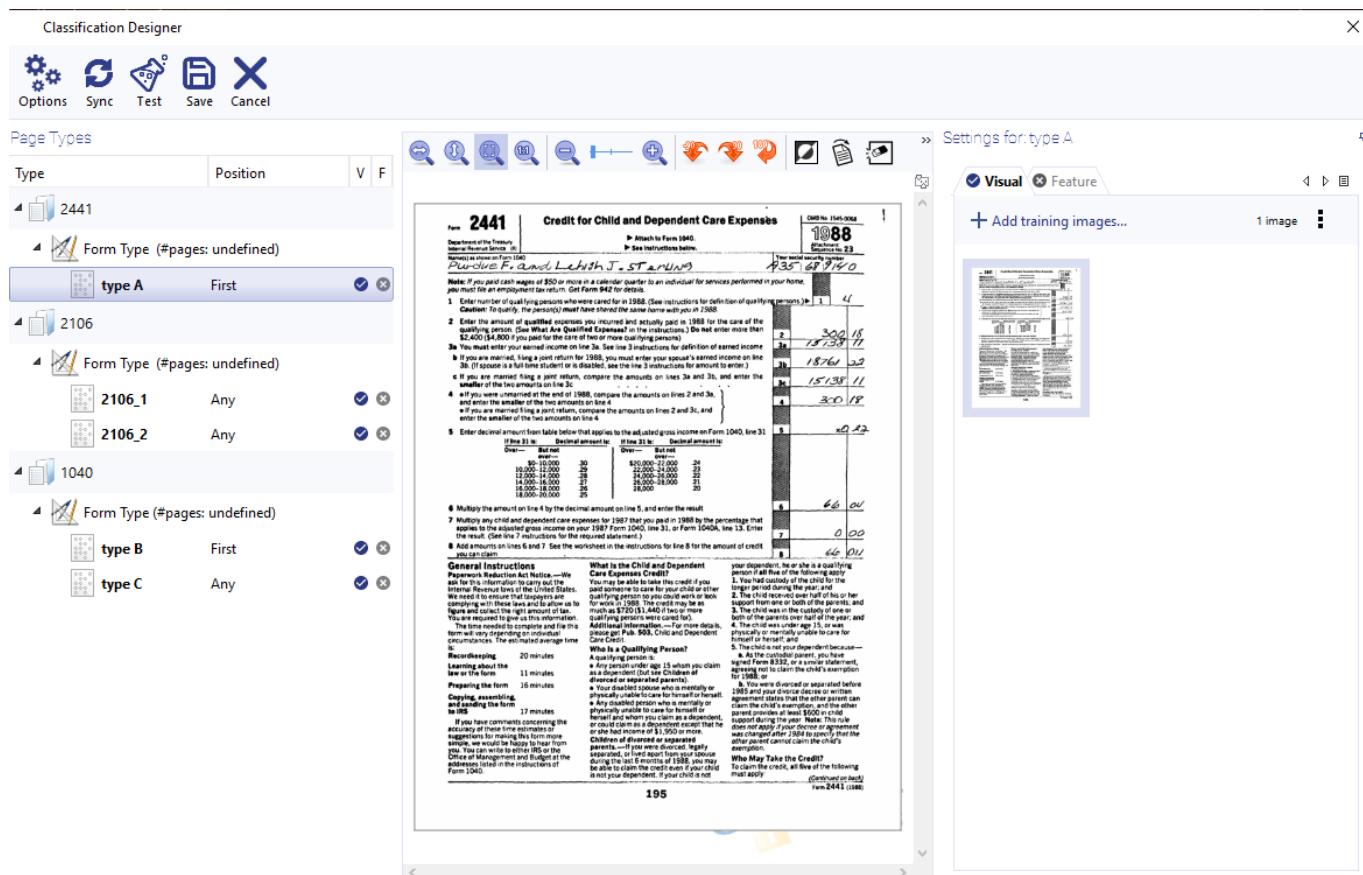
Adding and configuring classifiers for the Page types

Each Page Type represents an image class in the scope of classification. The goal is for each processed image to be mapped with one of the available Page Types, or remain unclassified if it is not a part of those categories. The classification process is handled by the classifiers, specifically two types: the Visual Classifier and the Feature Classifier. Each classifier type can be distinctly configured for each Page Type and should be done so depending on the Page Type case.

Visual Classifier

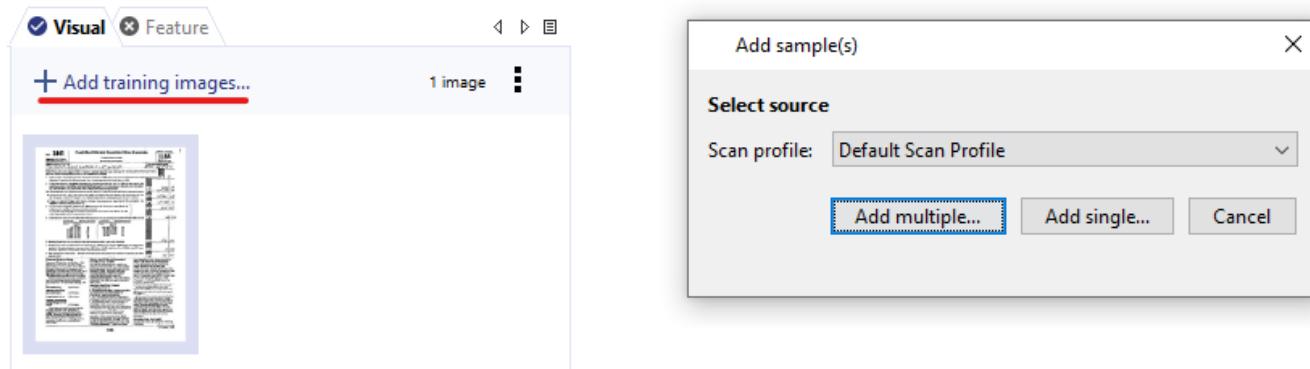
The Visual Classifier performs a visual comparison of the image. Although robust to small changes in the image, dealing mostly with form-like images means that each image will be different at least with regards to the filled in content. Even though the content itself does not take part in the comparison, its structure does. In a multi-line field for example it doesn't matter what is written but how much is written. Taking that into account, achieving the best results with a Visual Classifier requires a large sample of different versions of the image. This translates as experience for the classifier, essentially exposing it to many different possible image structures that are all part of a specific Page Type category. The more different images it is aware of, the higher the probability that a new incoming image will be similar to one of the already known ones, resulting in successful categorization. The different image versions are known as Training Images and the digestion of those images by the Visual Classifier and the experience build-up is known as Training.

In order to provide Training Images to a Page Type classifier, select it and on the right panel select the Visual Classifier tab. Note that the Visual Classifier should already have one training image which matches the Sample Page defined in the Form Type.



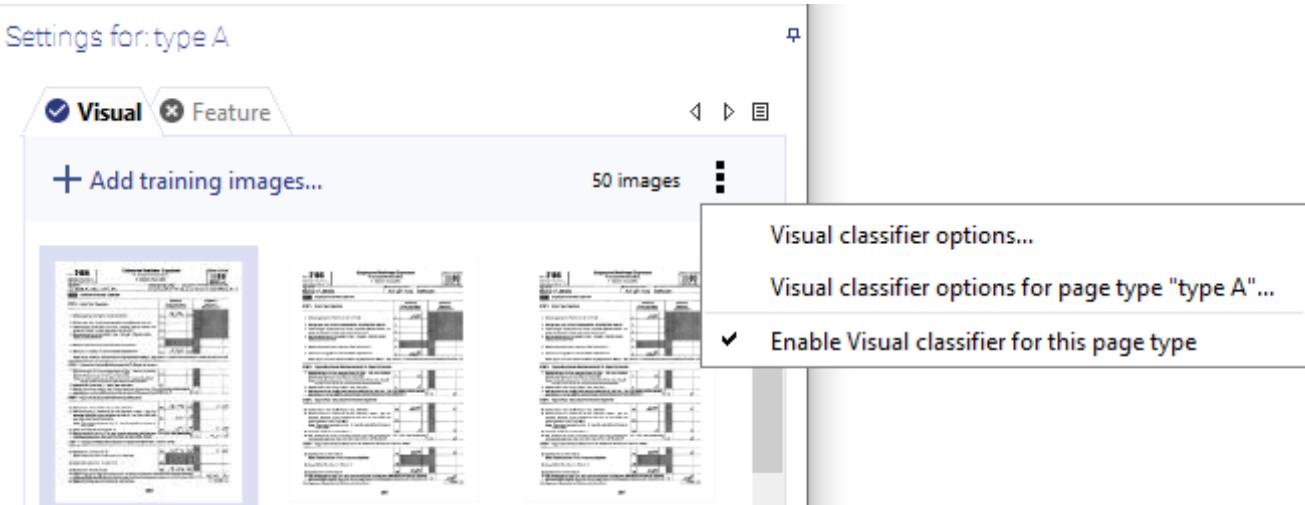
With the Visual Classifier tab active, press the **+Add training images...** in order to open the file system browser.

Settings for type A



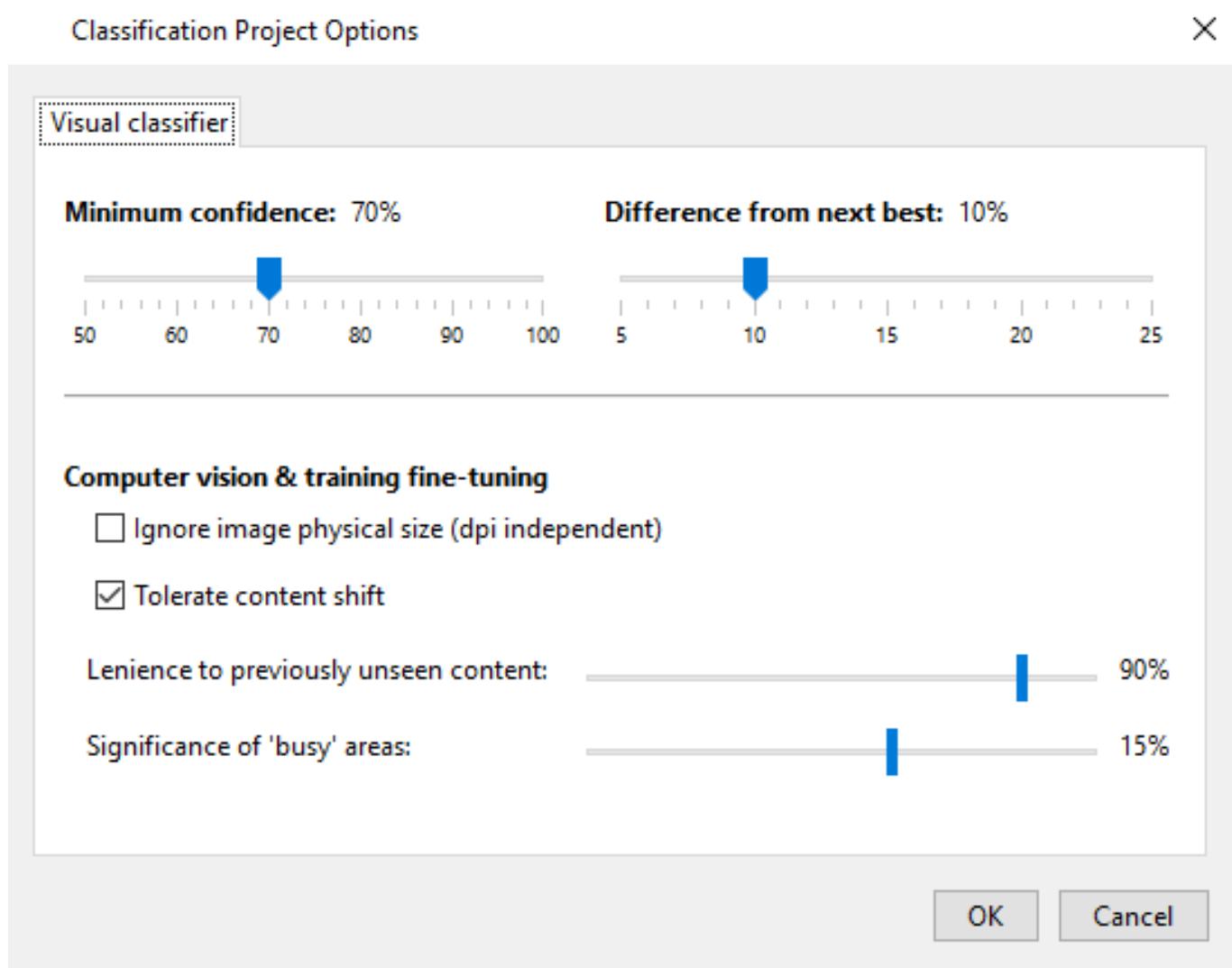
Although there is no enforced minimum or maximum amount of training images, in order to achieve good results it is recommended to provide at least 10 marginally different versions of the Page Type image. Providing more versions will increase the classification success rate. Providing too many images however has diminishing returns and will potentially make the classification process slower. A good threshold is around 50 versions of the image but the best results will be achieved via trial and error by testing the classification project using the gathered Test Set and iterating on its configuration depending on the test results.

Once the Training Images have been provided to the Visual Classifier, there is one last configuration that should be looked at, namely the Visual Classifier options. By pressing the sandwich button on the right of the training image count, the classifier options context menu appears.



Visual classifier options

The first option opens a dialog that offers common configurations for all the Visual Classifiers of all the defined Page Types. The selected values will govern the results of the classification project.



Basic options:

1. Minimum confidence

The minimum confidence value is the cut-off similarity threshold which decides if a processed image actually is a match with a Page Type category or not. When the Visual Classifier compares a processed image with a Page Type training data, the result is a percentage value named similarity confidence i.e. how similar the image is with the Training Images that represent that specific Page Type. Although it would be best to achieve similarity levels in the 95%, if the images have multiple variations between themselves these levels usually drop to the 80 or even 70% range.

Setting the minimum confidence at 70% for example essentially means: If an image is at least 70% similar to a Page Type, consider that Page Type as a category candidate for that image.

A good default value for the minimum confidence level is 70% however this is heavily dependent on

the content that will be processed by the Classification Project and will be calibrated based on the Testing iterations.

2. Difference from next best

It is very likely that a processed image will get similarity levels over 70% with multiple Page Types. This results in the classifier being unable to decide which Page Type to assign to the processed page leaving it as unclassified. For this reason there is another type of confidence threshold, the difference from next best. The assigned value dictates the minimum confidence difference the best and second-best results must have in order for the best result to be considered as a proper match. For example, assuming 70% minimum confidence and 10% difference from next best, if Page Type A has 90% confidence and Page Type B has 78% confidence, although both are over the minimum confidence threshold, the processed image will be classified as Page Type A because the confidence differences are larger than the configured 10% ($90 - 78 > 10$). On the other hand, if Page Type B had achieved 85% confidence then the processed image would remain unclassified due to it being too similar to both the Page Type A and Page Type B categories. It is important to note that the difference from next best must be satisfied even if the second-best result is below the minimum confidence threshold.

Advanced options:

The advanced computer vision & fine tuning options offer some extra configuration that might help increase the Classification Project effectiveness, depending on the handled content.

1. Ignore image physical size (dpi independent)

Since visual classification is a visual comparison of two images, one of the attributes that are taken into consideration is the image physical size. Two similar images one scanned at 150 dpi and one at 300 dpi will not produce a match. In general, it is expected that the images scanned and imported into the system will have the same scan settings applied to them (one of them being the scan DPI). It is however possible to not only handle images with different DPI amongst themselves but also to handle images with no DPI information, as is the case with images that are produced from smart phone cameras. For such cases, enabling the DPI independent flag will allow proper handling of the content.

Note that results with the DPI independent flag will differ from the results without it, even when handling the exact same content. Additionally, this option state also produces different training data for the Visual Classifiers. This means that triggering the option on or off will require retraining the Classification Project in order to properly take effect.

2. Tolerate content shift

Although it is best to strive for good quality scanning, minimizing artifacts and distortions, it is very

likely that at least some shift will be present in the scanned images. Although minimal (e.g. 1-2mm) shifts are tolerated by the Visual Classifier, if larger shifts are expected then they must be taken into account. Although it would seem intuitive to have this as the default functionality, increasing content shift tolerance also increases the time the classification process takes to complete. It is thus left in the discretion of the Classification Project administrator to have it enabled or not.

3. Lenience to previously unseen content

During training, the Visual Classifier generates training data based on the Training Images available to it. This training data contains the features that characterize the specific Page Type. During classification, the Visual Classifier attempts to find those extracted features in the image being processed and the similarity confidence depends on how many it finds. Being dependent on the Training Images, the training data features could potentially be incomplete. In such a case and when the system is presented with a different version of the Page Type that contains different features, the similarity confidence will drop. The Lenience to previously unseen content value essentially affects the generated training data in such a way that features missing from the training set of images could potentially be discovered during training, thus covering cases such as the one described above.

Higher values on this configuration generate training that fits the provided training set, i.e. the generated features are closely tied to the available training images.

Lower values on this configuration generate training data that fits less and is more of an interpolation between the provided training set images. This interpolation is what allows the discovery of potentially "in-between" features of two training images. The negative impact of low options values however is that the known, at-hand features of the training set are not exactly taken into account which means that processed images that bear the same features will show lower similarity confidence scores even though they almost perfectly match the training set images.

4. Significance of 'busy' areas

Busy areas in an image, i.e. areas where high density content lies, are also the areas where the image features are present. Imagining 2 blank images, one would treat them as the exact same. However, adding just one line of text at the top of image A and one line of text at the bottom of image B, they suddenly become different and one would potentially assign different Page Type categories to them. This example, although lacking real use case realism, goes to show that it is the content that drives the Visual Classification process.

The Significance of 'busy' areas value modifies the weight that is given to these parts of the image when they are compared between the processed image and the training data.

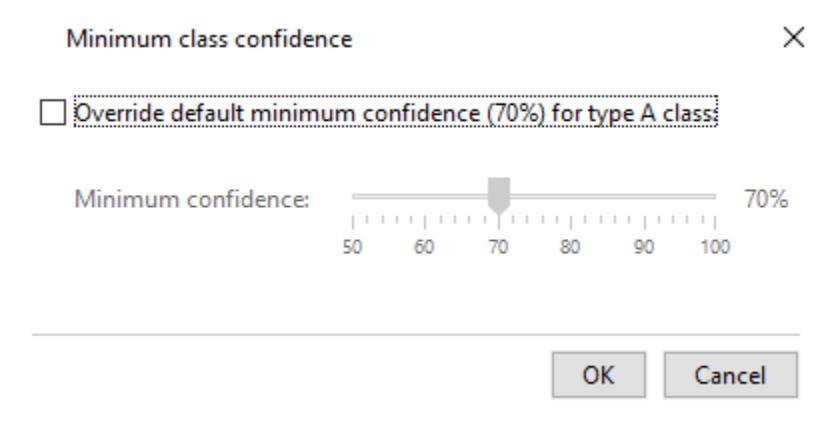
Higher values on this configuration give higher similarity confidence if the busy areas are similar. Otherwise, if the busy areas between the processed image and the training data have differences

then the similarity score decreases, even their overall content density is the same.

Lower values on this configuration lower the importance of content similarity between the busy areas and instead favor similar content density.

Visual classifier options for page type

The second option on the Visual Classifier context menu pertains to the specific Page Type handled by the classifier.



Only one option is available here and it is in the form of a Page Type specific override of the common minimum confidence value configured for the Visual Classifier. If configured, the common minimum confidence value no longer applies to the specific Page Type. In order for a processed image to be classified as that specific Page Type its similarity confidence value for that type must be over the newly defined confidence threshold.

For example, assuming Page Type A and Page Type B, with a Visual Classifier minimum confidence of 70% and a Page Type A confidence override of 80%, a processed image that gets 78% similarity with Page Type A and 67% similarity with Page Type B will essentially remain unclassified.

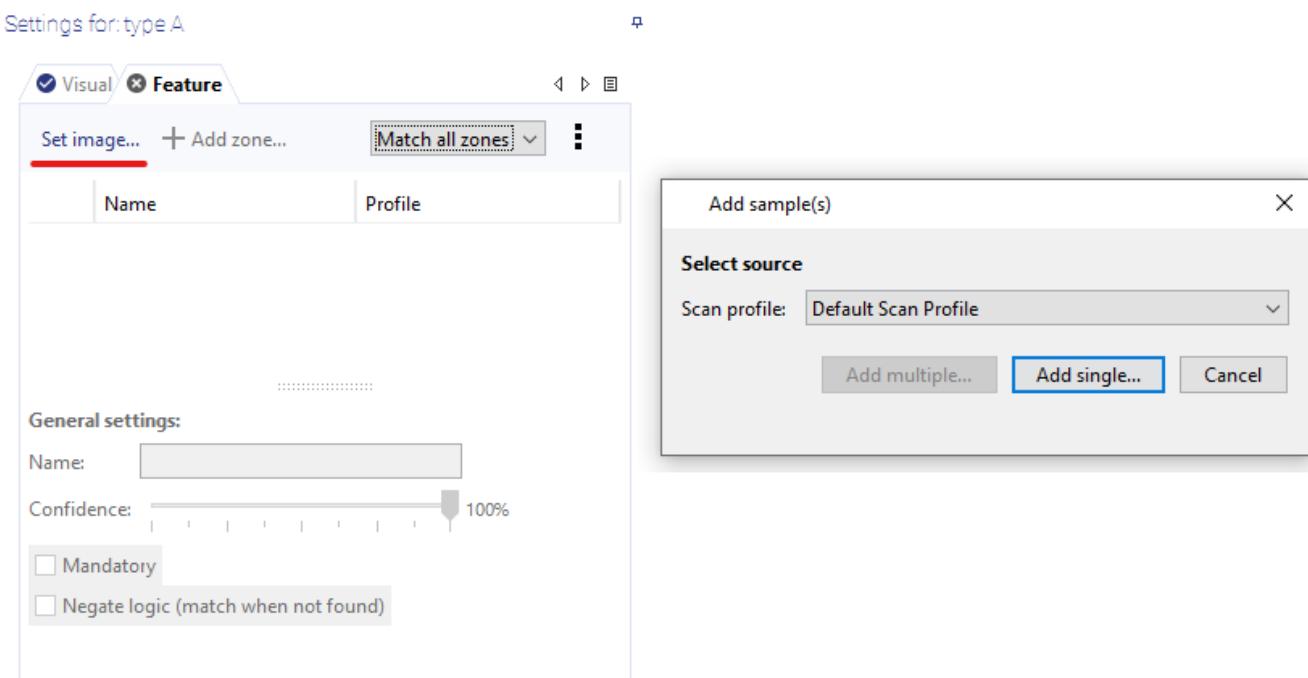
The override option allows for better fine tuning of the Classification Project, enabling stricter rules to apply for specific Page Types that have lower tolerance for misclassifications.

Feature Classifier

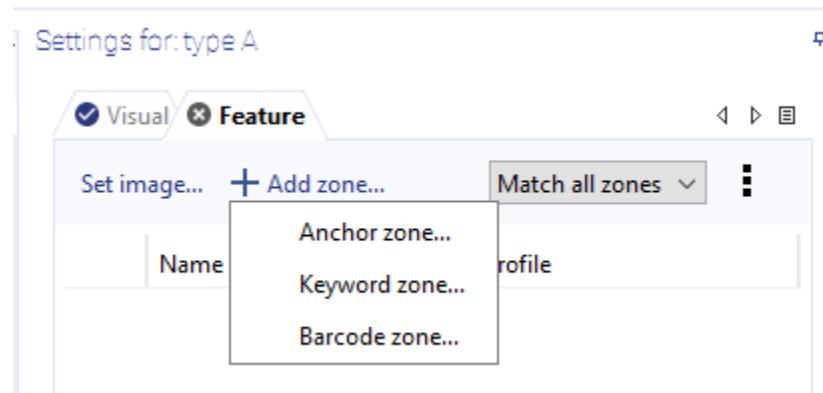
The Feature Classifier attempts to detect manually defined features in the image. The feature definition is done on a template image that is also manually defined. The Feature Classifier can only work when a template image is defined and only one image is supported. Because the selected features will have to be present on all the processed images of that Page Type, it is important that their selection must not comprise variable features like filled in checkboxes or other types of input fields. Instead, static information like field labels (that will be the same regardless of the way the form is filled), logos and other meta-information such as the word Page in the bottom right "Page 3 of 5" (if applicable) are best as

selected features.

In order to enable the Feature Classifier for a Page Type, a template image must first be defined. By selecting the Feature tab, press on Set image... and add a single image.



As explained above, it is advised that the selected image is a clean, unfilled scan of the form that serves as the template from which all other versions will be constructed. Once the image is selected, zones must be defined next. A Feature Classifier zone is a rectangle inside which an image feature resides. In order to define a zone, press the + Add zone...



and select a type of zone from the drop down menu. The selected type will depend on the underlying part of the image that will be selected inside the zone rectangle and the type of comparison mechanism that will be employed when attempting to detect the feature in the zone in the processed images.

Matched zones and classification result

For the Feature Classifier to assert a Page Type match, it has to successfully detect the provided features on the template image in the processed image. Many times however it is not mandatory for all features to be present in the image, or some features might fail to be detected for various reasons. To account for this, classification can optionally succeed even if a subset of the features are available. This can be configured from the dropdown menu on the right of the Feature classifier tab. The available options depend on the defined feature zones.

Settings for: type A

Match all zones

- Match all zones
- At least 1 zone(s)
- At least 2 zone(s)
- At least 3 zone(s)

Name	
Zone #1	<input checked="" type="checkbox"/> <input checked="" type="radio"/>
Anchor #1	<input checked="" type="checkbox"/> <input checked="" type="radio"/>
Keyword #1	n/a

Zone General settings

For each defined zone, the following general settings can be configured:

General settings:	
Name:	Zone #1
Confidence:	<input type="range" value="90"/> 90%
<input type="checkbox"/> Mandatory <input type="checkbox"/> Negate logic (match when not found)	

Name

The zone name can be any string and helps identify the zone with a quick glance.

Confidence

Although feature detection ends up being a binary assertion (exists or doesn't exist), the detection process itself does not always provide a binary result. Instead, there is a confidence metric that can be configured to a specific threshold. If the feature match confidence is above that threshold then the feature is considered as existing in the processed image.

Mandatory flag

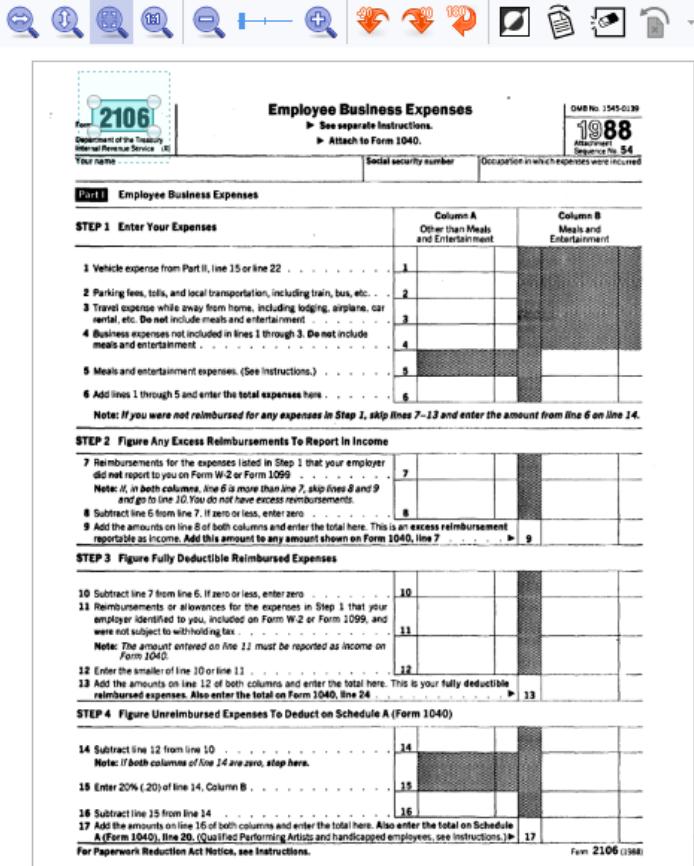
Although the classifier can assert a Page Type match without all zones being present in the processed image, it is possible to declare some zones as mandatory. This means that even if enough zones are found during detection, if a mandatory zone is not part of the set then the classifier will not consider the Page Type.

Negate logic flag

This flag is the reverse of the Mandatory flag. If the feature is found then the classifier will not consider the Page Type.

Anchor zone

When defining an Anchor zone, the feature that will be used during the feature matching process is the image snippet inside the rectangle. The matching process is an image comparison matching where the snippet from the template image will be compared with the snippet under the same area in the processed image. If the snippets under the two areas match then the feature is considered as existing in the processed image.



Employee Business Expenses
See separate instructions.
Attach to Form 1040.

OMB No. 1545-0139
1988
Sequence No. 54

Part I Employee Business Expenses

STEP 1 Enter Your Expenses

	Column A Other than Meals and Entertainment	Column B Meals and Entertainment
1		
2		
3		
4		
5		
6		

STEP 2 Figure Any Excess Reimbursements To Report in Income

	Column A	Column B
7		
8		
9		

STEP 3 Figure Fully Deductible Reimbursed Expenses

	Column A	Column B
10		
11		
12		
13		

STEP 4 Figure Unreimbursed Expenses To Deduct on Schedule A (Form 1040)

	Column A	Column B
14		
15		
16		
17		

Form 2106 (1988)
For Paperwork Reduction Act Notice, see Instructions.

189

Settings for: type A

Visual Feature

Set image... Match all zones

Name	Profile
<input checked="" type="checkbox"/> Anchor #1	Anchor Recognition Engine

General settings:

Name: Anchor #1

Confidence: 90%

Mandatory

Negate logic (match when not found)

Anchor settings:

Search zone width (in mm):

Search zone height (in mm):

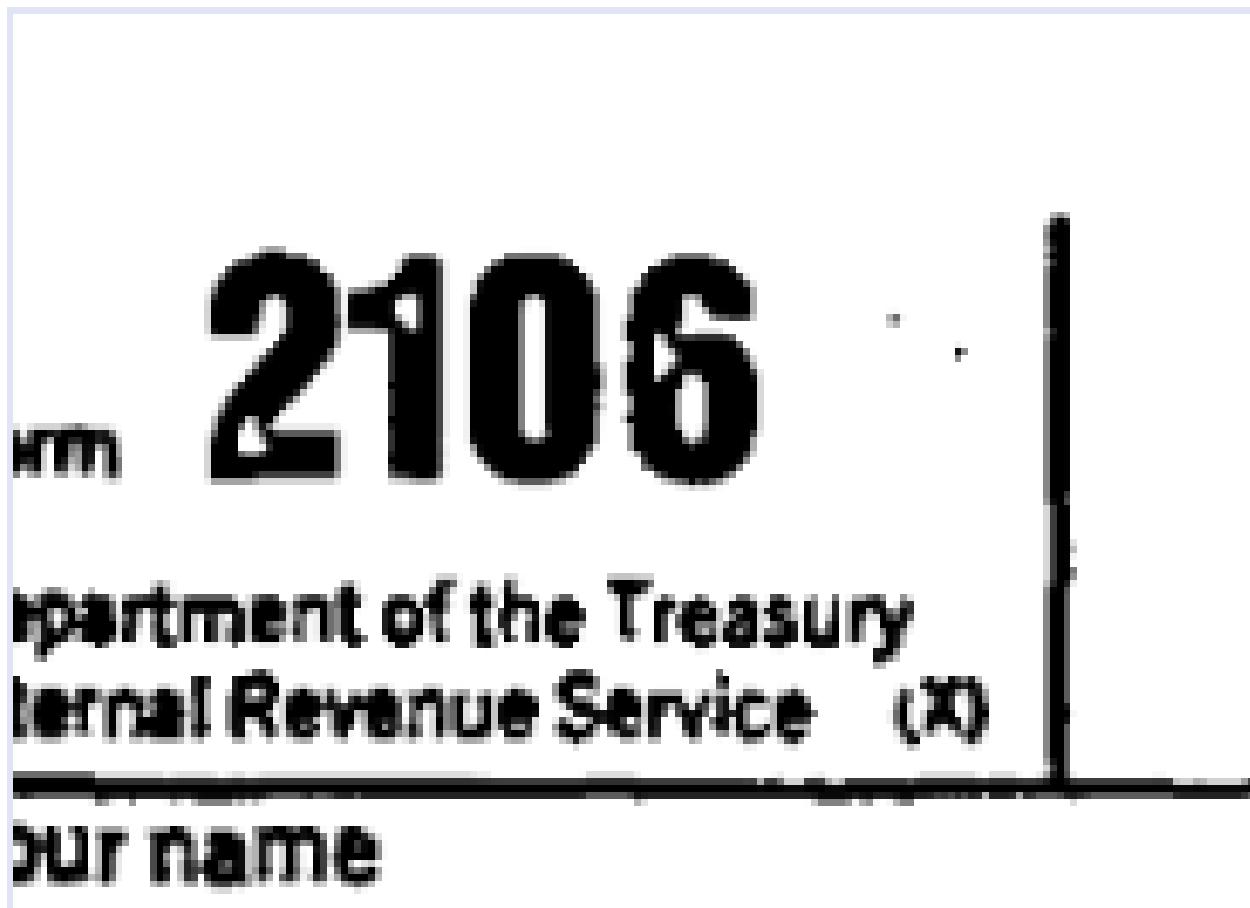


Enlarged image search pattern

Notice the type specific Anchor settings. They allow the definition of the search zone width and height. Due to possible image deformations like skews, off-center scans or other, it is very likely that the zone coordinates defined on the template image do not reference the same exact part of the processed image. For example, if the processed image was scanned off-center, then the same zone coordinates might contain the following image snippet:



Which will not match very well with the intended feature. For this reason, a search zone is defined which is a rectangle that encompasses the anchor zone, within which detection of the image snippet feature will be attempted. In this case, the search zone will contain the following part of the image:



which clearly contains the original image snippet feature. A balance must be stricken when deciding on a search zone size. A large search zone will almost certainly account for all deformations and thus allow detection of the feature, however it will also increase processing time. On the other hand, a very small search zone will allow for quick feature detection but it is likely that scan deformations will place the feature outside of its reach, making detection impossible. Experimentation with a good training and test set of images will guide optimal zone definition.



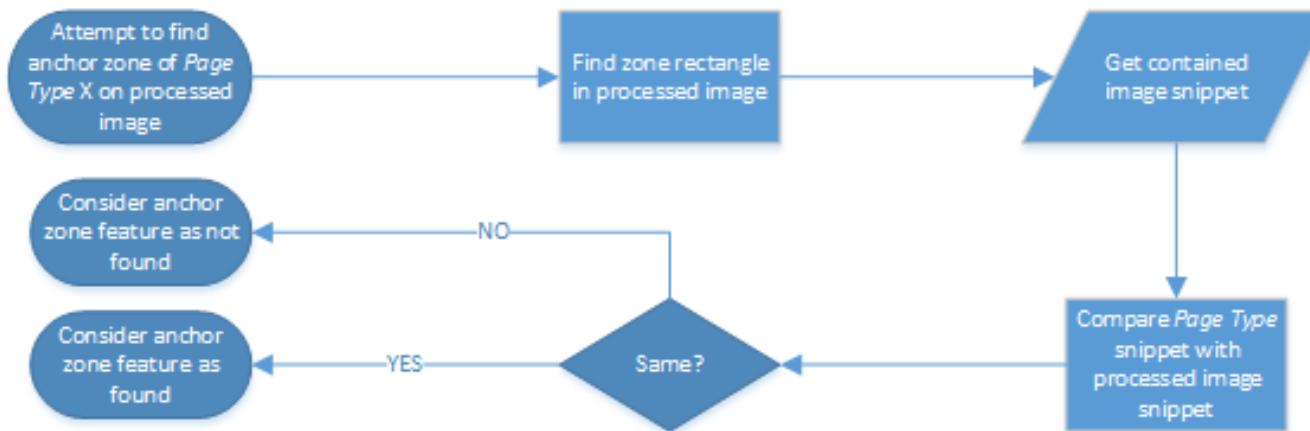
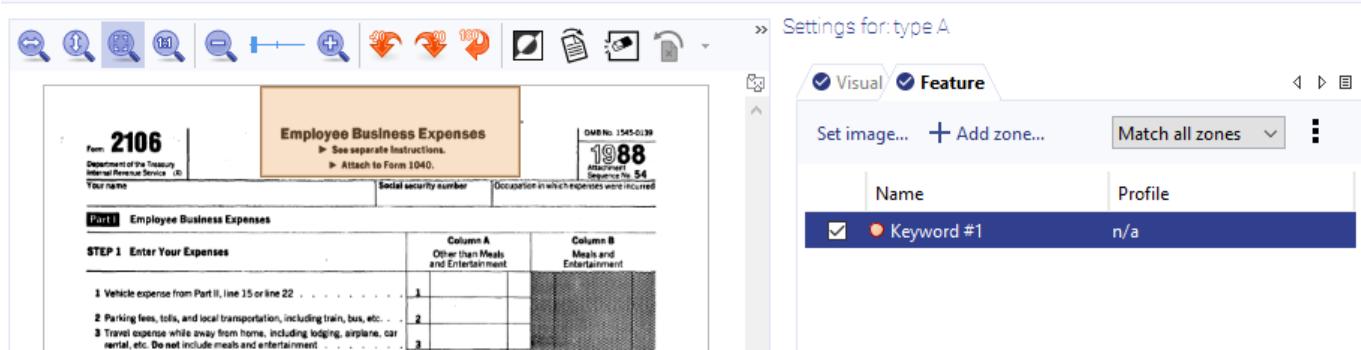


Figure 49. Anchor zone detection on processed image

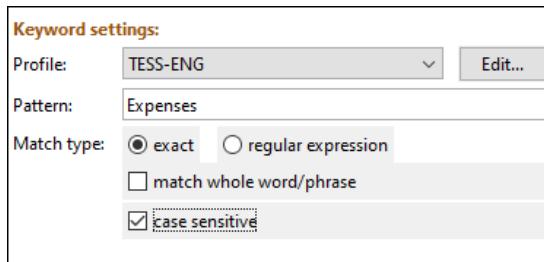
Keyword zone

When defining a Keyword zone, the feature that is used is the OCR text of the image snippet inside the rectangle. In the template image, only the zone rectangle is defined. During feature detection, the same rectangle in the processed image is extracted and goes through text OCR. The text result is compared with a predetermined text string for that specific zone. If the strings match, then the feature is considered as existing in the processed image. Note that there is no comparison done between the template and processed images. The template is only needed to define the zone rectangle.



When adding a Keyword zone rectangle, it is not mandatory to set the rectangle only around the specific keyword part. In contrast to Anchor zones, Keyword zones do not have search zones. This means that the zone itself must be large enough to account for possible scan deformations and although the OCR generated text will contain more than the target keyword, as long as the keyword is captured the feature is considered as present.

Under the Keyword Settings, there are three configurations:



Profile

A dropdown list with the OCR Extraction Profile that will be used for extracting the text from the zone image snippet. The list of available profiles depends on the already configured Extraction Profiles. The Edit... button is a shortcut for quick editing.

Pattern

The input field should contain a pattern string which represents the target keyword we are looking for in the extracted text.

Match type

When attempting to detect the defined Pattern above, the detection can either be an exact match or a regular expression match. An exact match will attempt to find the provided Pattern in the OCR text result, whereas a regular expression match will apply the provided regular expression in the Pattern field into the OCR text result.

Exact match example:

Setting the Pattern as Expenses with the OCR text result being Employee Business Expenses will end up with a successful detection.

RegEx match example:

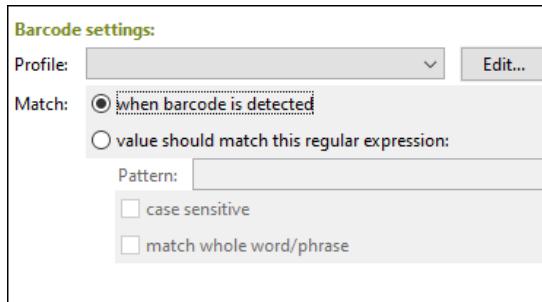
Setting the Pattern as Employee \w*? Expenses with the OCR text result being Employee Business Expenses will end up with a successful detection.

The Match type has two extra configurations, namely match whole word/phrase and case sensitive. The former will require that the Pattern mirrors the OCR text result, otherwise detection fails. This means that if the OCR text result contains more text than the Pattern then the match fails. The latter enforces case sensitivity to the match method. If unchecked, the pattern should be in lower case. Both these options allow for stricter matching rules.

Barcode zone

The Barcode zone is similar to the Keyword zone but instead of defining a rectangle containing text content, it contains a barcode snippet. Most of the logic for its detection is the same as the Keyword

zone except for the following configuration under the Barcode settings:



Match

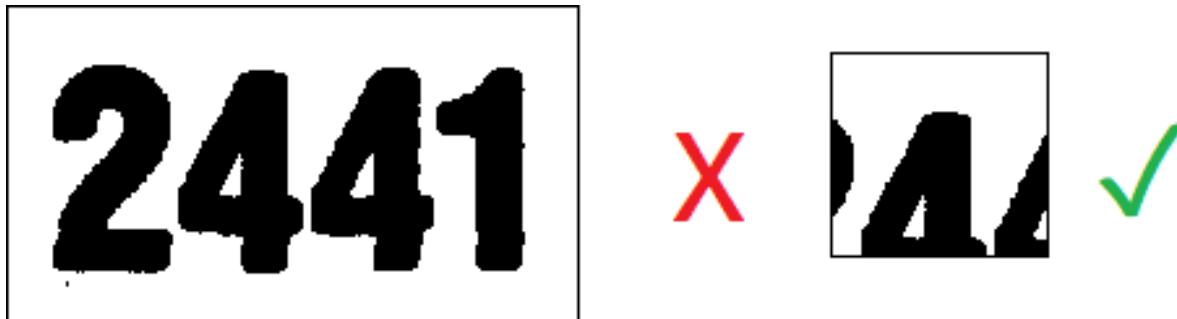
The Match configuration has two modes. It can either report the feature as existing if a barcode is found, regardless of its content, or it can additionally require the barcode content to match some Pattern criteria, akin to a Keyword zone match. As a reminder, the barcode type that will be detected is defined by the barcode recognition profile that is selected in the Profile dropdown.

Optimal zones

It is important to keep in mind the characteristics of an optimally defined zone in order to achieve high classification rates with low processing times. Although there are some guidelines for each zone type, the choice for the best zones will be influenced not only by the template image for the Page Type but also by the rest of the Page Types in the project. Naturally, choosing similar zones on different Page Types should be avoided since it can potentially confuse the Feature classifier decision process resulting in misclassification. Additionally, choosing zones in Page Type A such that they can also be identified in other Page Types is also problematic, again because they will be identified in processed images of Page Type B for example resulting in misclassification. Great care must be taken to analyze not only the template images for each Page Type but also the potential versions of the processed images in order to ensure that no cross Page Type features are defined.

Optimal Anchor zones

Although it might sound counter-intuitive at first, anchor zones should contain small patterns of pixels that are deemed unique to the Page Type, when compared to the rest of the images that will be processed. This retains the uniqueness characteristic of the feature whilst greatly increasing its detection speed. A logo on the top left of a form is probably going to be unique to that image. A fraction of that logo is still going to be unique to that image. There is no reason to look for the whole logo when we can look for a small part of it.



Notice that the selected zone doesn't contain a single character, since the number 4 or 2 could possibly be detected in other forms such as the 1040 form. However the combination of two partial 4 characters and a fraction of the 2 character is unique to the 2441 form, in the case of our Classification project.

Depending on the rate and severity of the scan deformations that occur during production scanning, the configured search zone must be large enough in order to allow detection of the anchor zone snippet. The performance of the feature detection is dependent not only on the snippet size but also on the search zone size which must be kept in moderation if possible.

Optimal Keyword zones

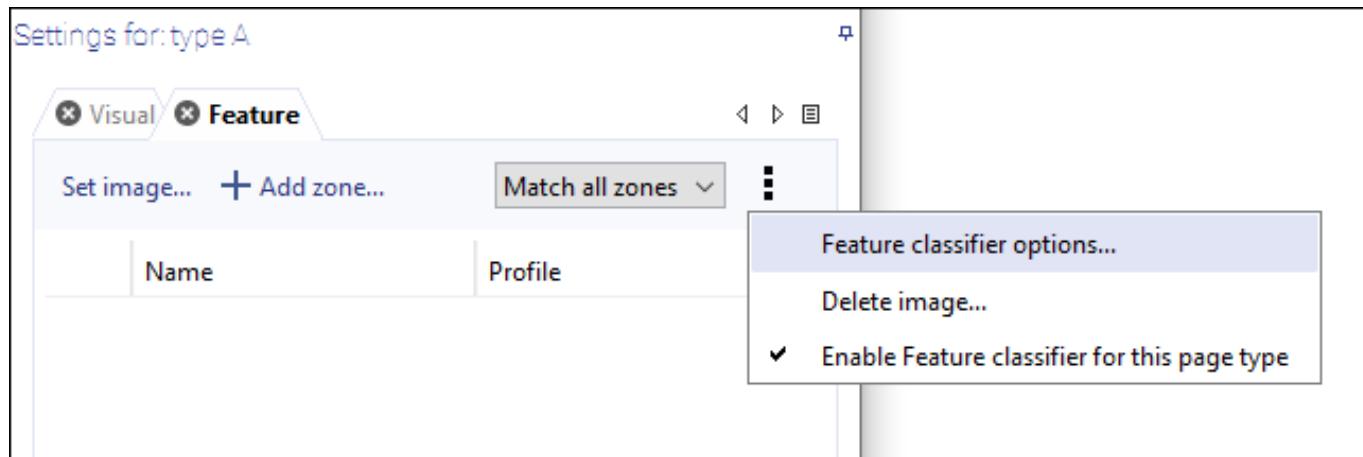
When defining a Keyword zone two characteristics are important, the zone content uniqueness and the extraction difficulty of the contained text. The text content itself should not affect the feature detection, however the recognition profile used to extract it, could potentially return partial or deformed OCR results. Characters such as I can be mistaken for the number 1 or O for 0 etc. Such Keyword zones should be identified and if used, the match confidence percentage should be configured accordingly.

Optimal Barcode zones

Regarding Barcode zones, the selection is relatively simple. The zone rectangle must contain the entirety of the barcode while being large enough in order to account for possible scan deformations that might occur. Obviously, accounting for barcode content will be mandatory if the same barcode type is used across different Page Types.

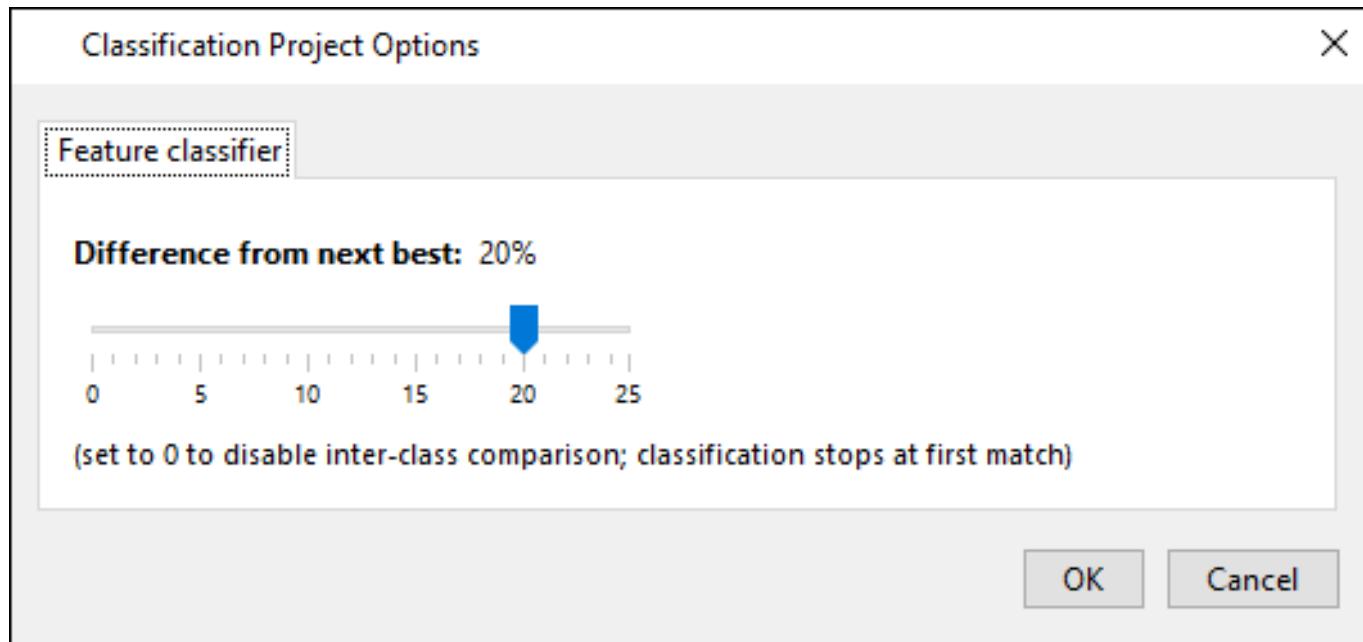
Feature classifier options...

Similar to the Visual classifier, the Feature classifier has generic configurations, accessible from the menu on the right.



Difference from next best

Similar configuration to one available to the Visual classifier. The Feature classifier also generates a confidence number for each Page Type, derived from the confidences of the detected features. If multiple Page Types have been designated as classification candidates, the difference from next best threshold can potentially help the Feature classifier decide which is the true Page Type of the processed image whilst guarding from misclassifications.



Page Types

On the Page Types left pane, each Form Type has its respective Page Types defined under it.

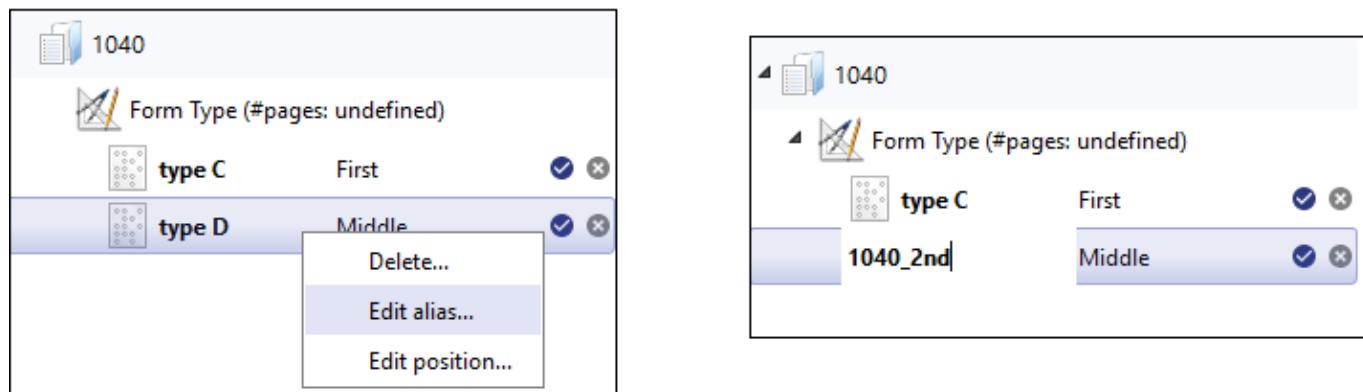
Page Types

Type	Position	V	F
2441			
Form Type (#pages: undefined)			
type E	First	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2106			
Form Type (#pages: undefined)			
type A	Any	<input checked="" type="checkbox"/>	<input type="checkbox"/>
type B	Last	<input checked="" type="checkbox"/>	<input type="checkbox"/>
1040			
Form Type (#pages: undefined)			
type C	First	<input checked="" type="checkbox"/>	<input type="checkbox"/>
type D	Middle	<input checked="" type="checkbox"/>	<input type="checkbox"/>

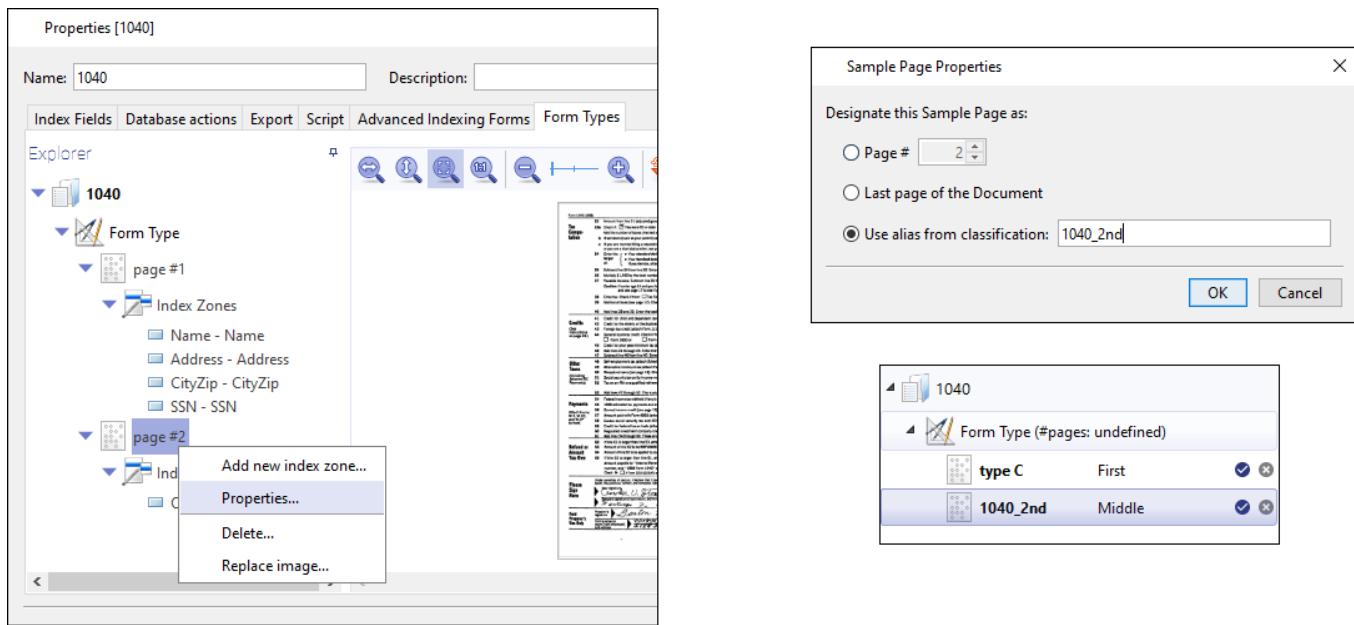
Each row under the Form Type represents a Page Type and each one has the following values associated with it:

Type alias

Since the Classification project is a standalone entity, it is not aware of the indexing configuration in the Job and vice versa, allowing Form Types and Classification projects to be created and maintained independently. Despite that fact, during runtime they still need to be combined since classification generates information that helps properly execute information extraction and indexing. The Classification project matches the Document Pages with Page Types and proceeds to identify the correct Form Type. Once the Form Type is selected and the workflow proceeds to automatic extraction, a map between the Document scanned Pages and the Form Type Sample Pages is needed, otherwise it is impossible to know which Index Zones to extract from which scanned Page. One way to achieve this mapping is in the form of the type alias which matches the discovered Page Type to its respective Sample Page. The type alias is a name that represents a specific Page Type in the Classification project as well as the rest of the system. It is an editable field and can be given a string value of choice.



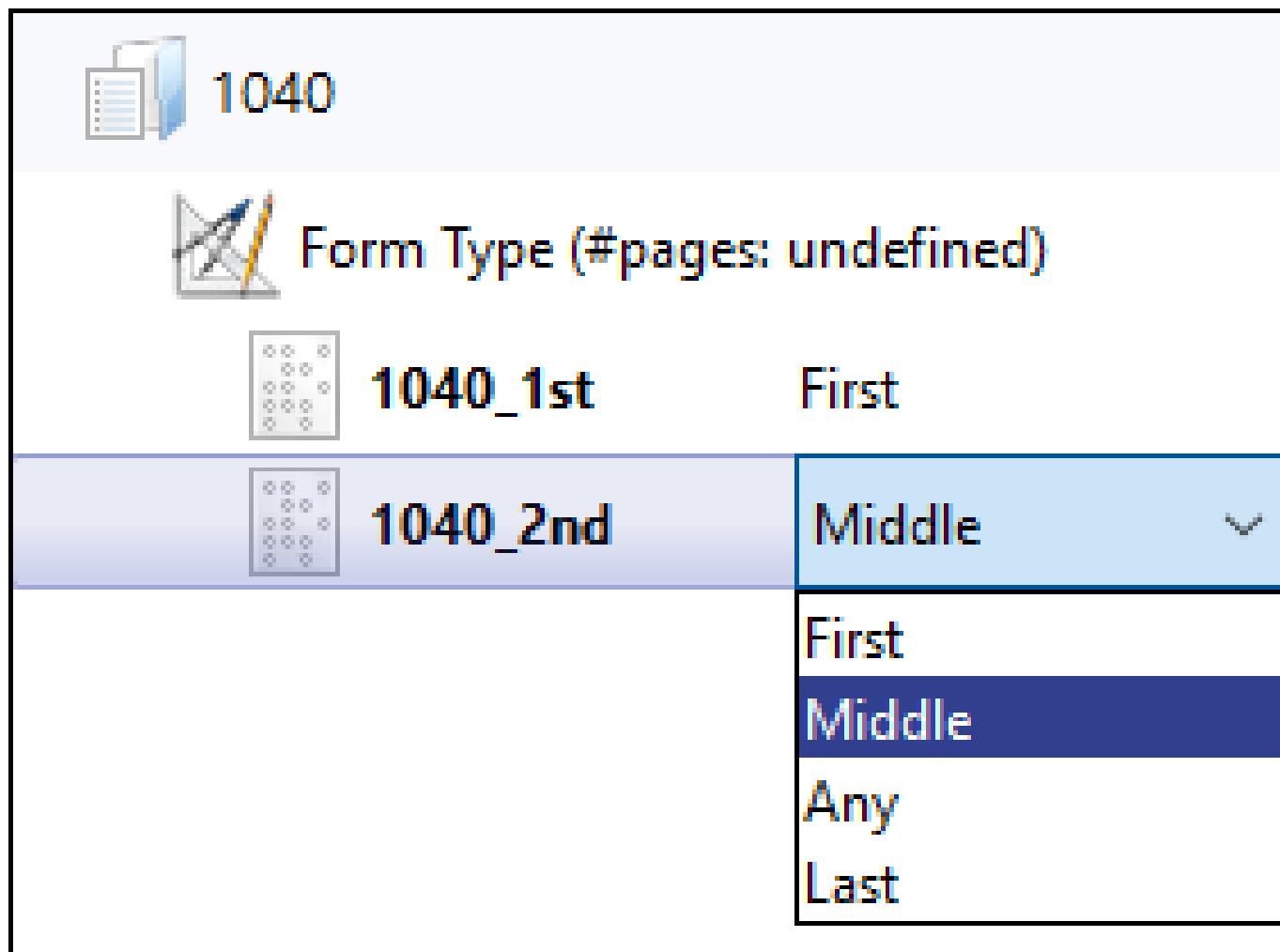
For the new alias to take effect in the Form Type, it must be manually defined in its respective Sample Page under the 1040 Document class Form Types definition.



Although mapping between Sample Pages and Page Types is mandatory for automatic extraction, it is not exclusively done via type aliases. Instead, the Sample Page positions can be used to the same effect, although this is not always possible.

Position

It is common for documents to be structured in a way such that some types of pages have static positions, usually first or last but not necessarily. A cover letter for example might always come at the beginning or the end of a job application. Another characteristic of documents is that their page-length could be variable. A CV for example has at least one page but its total content is dynamic. In order to accommodate such cases, 4 position types have been defined and can be assigned to each Page Type, namely First, Middle, Last and Any. Apart from being used for the mapping between Sample Pages and Page Types, their assigned values are what determines the result of the Form Type identification process. View the respective paragraph for further details.



First

This position should be given to a Page Type if it is expected to always be at the start of the document. Additionally, any Document Page that is classified as a First positioned Page Type will trigger a document separation if applicable. View the separation paragraph for further details.

Last

Similarly to the First position, any Page Type that is always expected to appear last should be assigned the Last position. Pages classified as a Last positioned Page Type also trigger document separation.

Middle

Any Page Type that will never appear at the beginning or at the end of a document should be given the Middle position value.

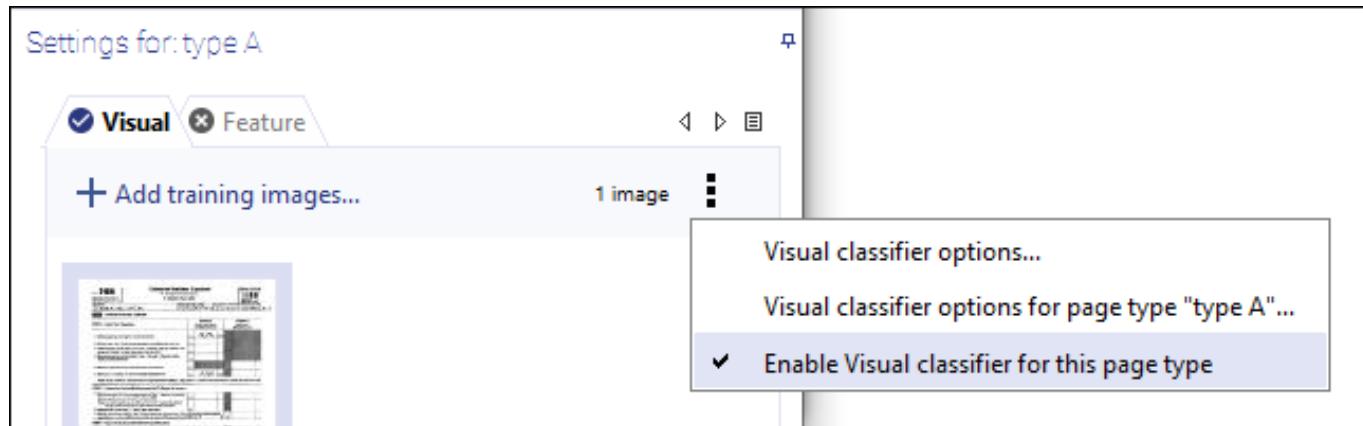
Any

Finally, if a Page Type can appear anywhere in the document, it should be given the Any position

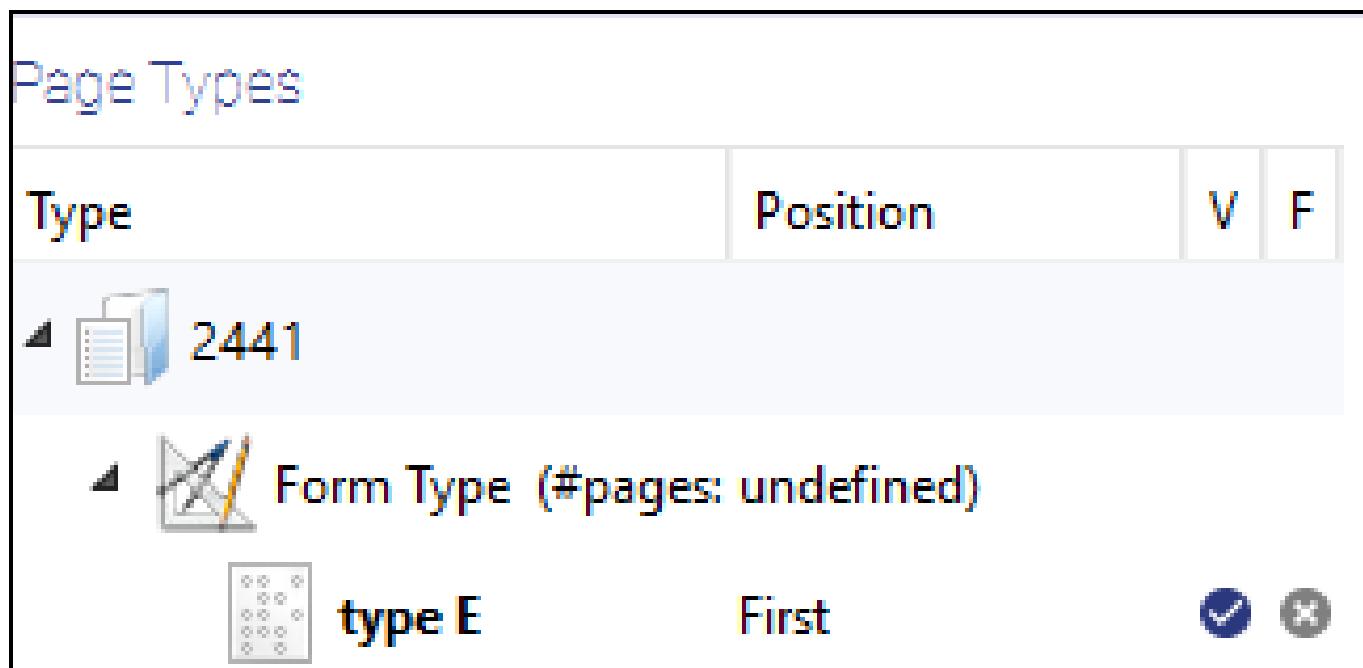
value. Note that if tagged as Any, no separation is triggered even if the page is positioned at the beginning or the end.

Classifier toggle states

Each classifier can be enabled or disabled for each Page Type respectively. This toggle action is available from the classifier menu on the right.



The toggle state is indicated for each Page Type for each classifier with a checkmark or X icon.

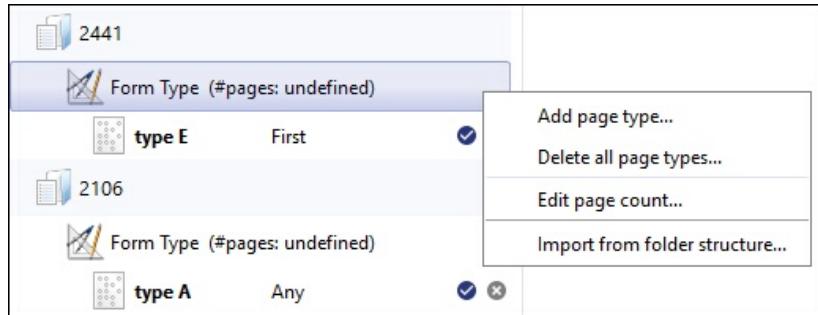


In this example, the (V)isual classifier is enabled whereas the (F)eature classifier is not.

Editing Form Type Page Types

As shown earlier, setting up a Classification project can begin by syncing it with the Job Document

classes. Although a big time saver, this is not mandatory. A Classification project will contain all the Document classes that are available to its Job, however their underlying Page Types are freely modifiable. By opening the Form Type context menu, it is possible to add new Page Types or remove all of them and start from scratch.



Another time saver is in the form of the *Import from folder structure...* option which offers the creation of multiple *Page Types* with *type aliases* based on the selected directory's folder names while importing each folder's image content as Visual Classifier training images.

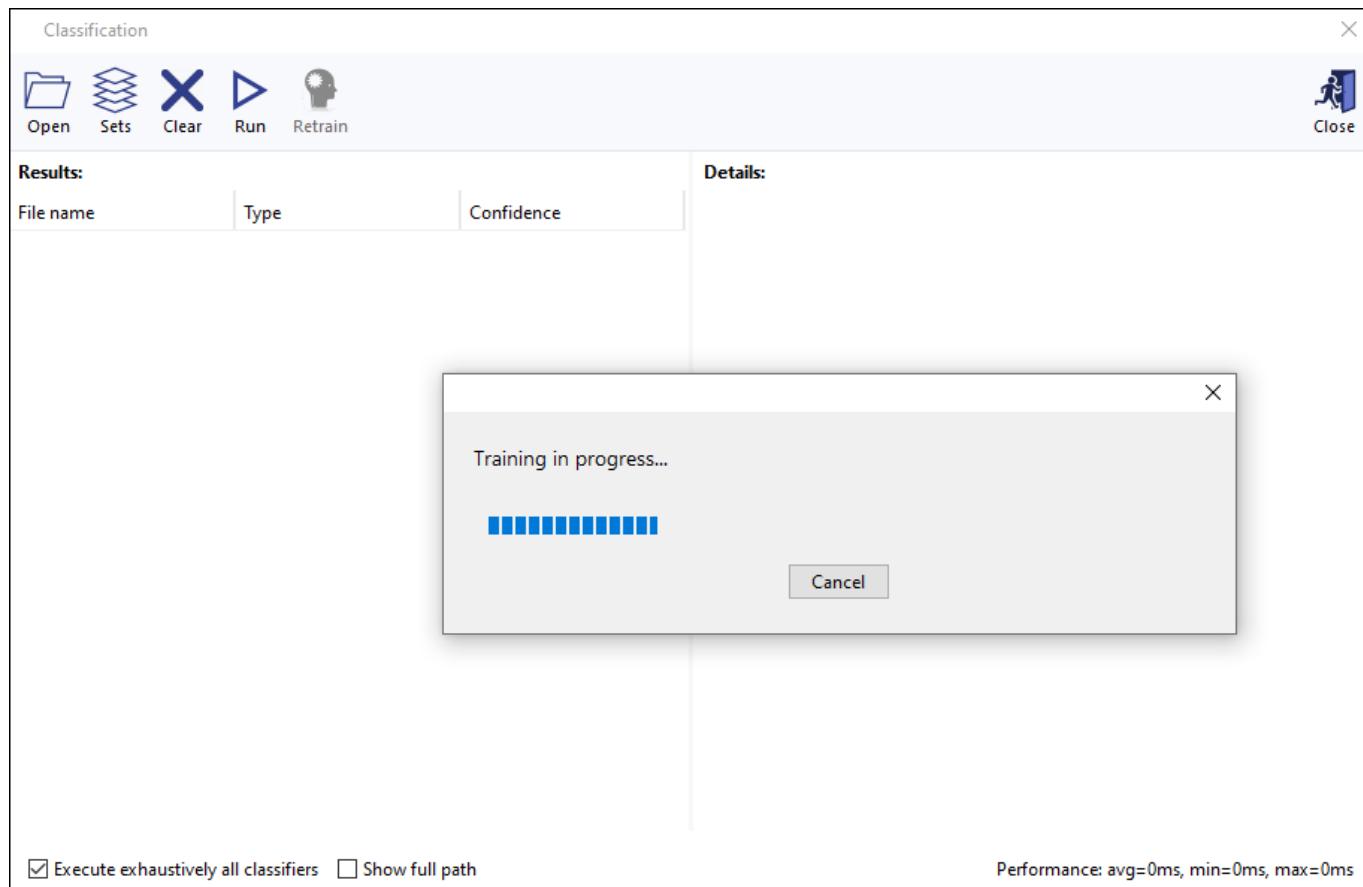


Finally, the *Edit page count...* option allows for the definition of a page count range for the specific *Form Type*, if known. This configuration provides better *Form Type* identification during the *Classify* step.

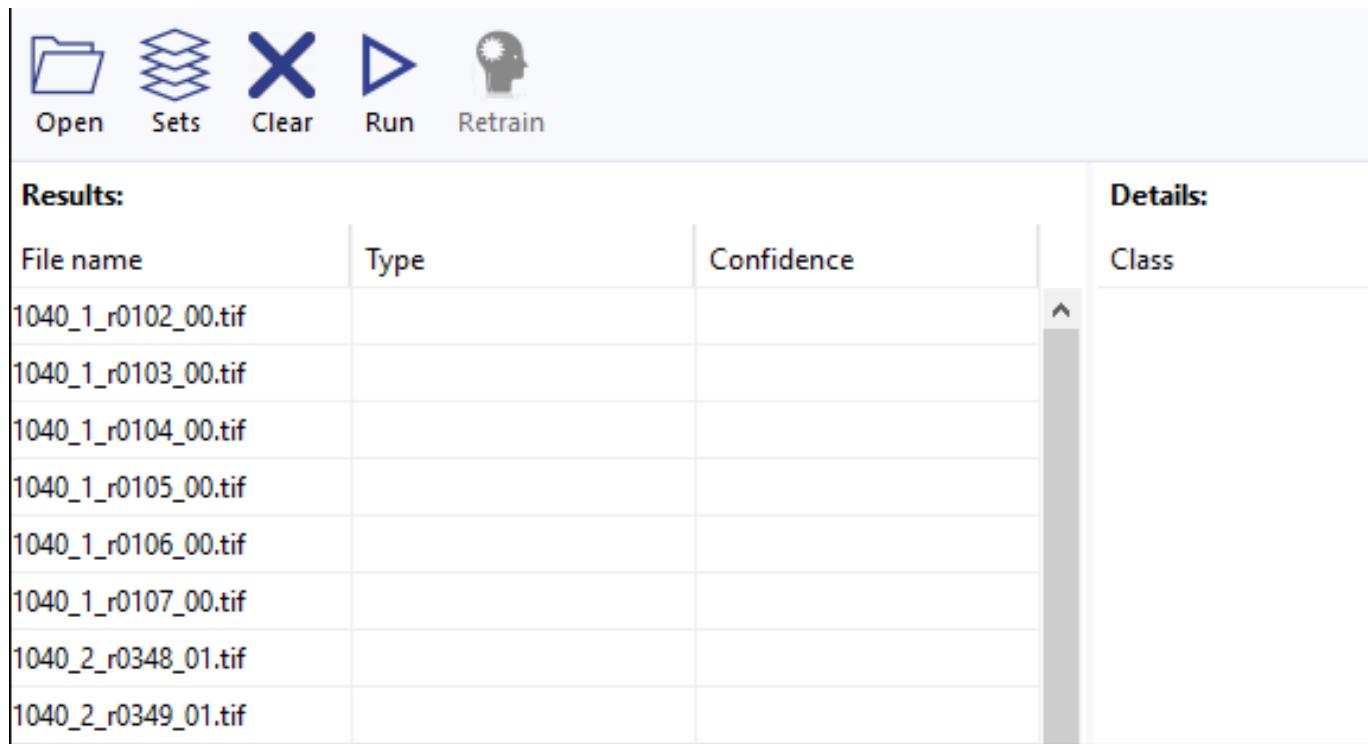


Testing and iterating on the project

Once the first iteration of the *Classification project* is ready, it is very important to begin a testing cycle in order to ascertain the success rate of the classification process. Open the Test dialog by pressing the *Test* button. This will also trigger the training process for the *Classification project*.



Once training is complete the *Classification* project is ready to be tested. As already explained, a large amount of the gathered image samples should be available for testing against the *Classification* configuration. To import the page samples, press the *Open* button. Once the test images are imported, they should be visible in the left panel of the Test dialog.



The screenshot shows the software interface with two main panels: 'Results' and 'Details'. The 'Results' panel on the left lists file names and their types. The 'Details' panel on the right shows classification confidence and class. A vertical scroll bar is visible on the right side of the 'Results' panel.

File name	Type	Confidence	Class
1040_1_r0102_00.tif			
1040_1_r0103_00.tif			
1040_1_r0104_00.tif			
1040_1_r0105_00.tif			
1040_1_r0106_00.tif			
1040_1_r0107_00.tif			
1040_2_r0348_01.tif			
1040_2_r0349_01.tif			

Once the test set is ready, you can start the Classification test by pressing the *Run* button.



The results appear on the result panel, showing the detected *Page Type* (under the *Type* column) and the Classification confidence (under the *Confidence* column). The green color means that the *Page Type* detection was successful but it does not necessarily mean that the detected type is also the correct one. This is where manual validation comes into play. It is suggested that the test image names and the *Page Type aliases* (reported in the *Type* column) should be such that it is easy to identify Classification correctness at a glance.



The screenshot shows the 'Results' panel with a single row of data. The file name is 4562_1_r0149_07.tif and the type is [FAIL]. A vertical scroll bar is visible on the right side of the panel.

4562_1_r0122_05.tif	[FAIL]
4562_1_r0132_06.tif	[FAIL]
4562_1_r0137_08.tif	[FAIL]
4562_1_r0140_05.tif	[FAIL]
4562_1_r0148_06.tif	[FAIL]
4562_1_r0149_07.tif	[FAIL]

Classification failures are marked with red color and [FAIL] type. In this example, the 4562 form was not part of the configured Form Types for this Classification project and thus, it properly failed to Classify as anyone of them. If it had been successfully classified as any of the existing Form Types, then this would be a misclassification and the configuration should be re-evaluated.

The right hand *Details* panel will contain the *Classification Confidence* results for each of the tested *Page Types*.

Results:			1040_1_r0104_00.tif		
File name	Type	Confidence	Class	Visual	Feature
1040_1_r0104_00.tif	1040_1	90.85	1040_1	90.85	100.00
1040_1_r0105_00.tif	1040_1	91.40	1040_2	56.10	34.59
1040_1_r0106_00.tif	1040_1	96.68	2441	52.69	33.85
1040_1_r0107_00.tif	1040_1	94.59	2106_2	51.93	38.04
1040_1_r0108_00.tif	1040_1	96.22	2106_1	51.55	41.61

For the image file **1040_1_r0104_00.tif** for example, the Visual classifier reported a 90.85% confidence that it is of the 1040_1 Page Type while the Feature classifier reported 100% confidence for the same Page Type. Since other Page Types had confidence scores low enough to be outside of the next best range defined for both classifiers, they are not considered as candidate Page Types and are marked with red.

It is possible to have classification fail for an image because the top two candidate Page Types have very close confidence scores (i.e. their difference is less than the configured difference from next best).

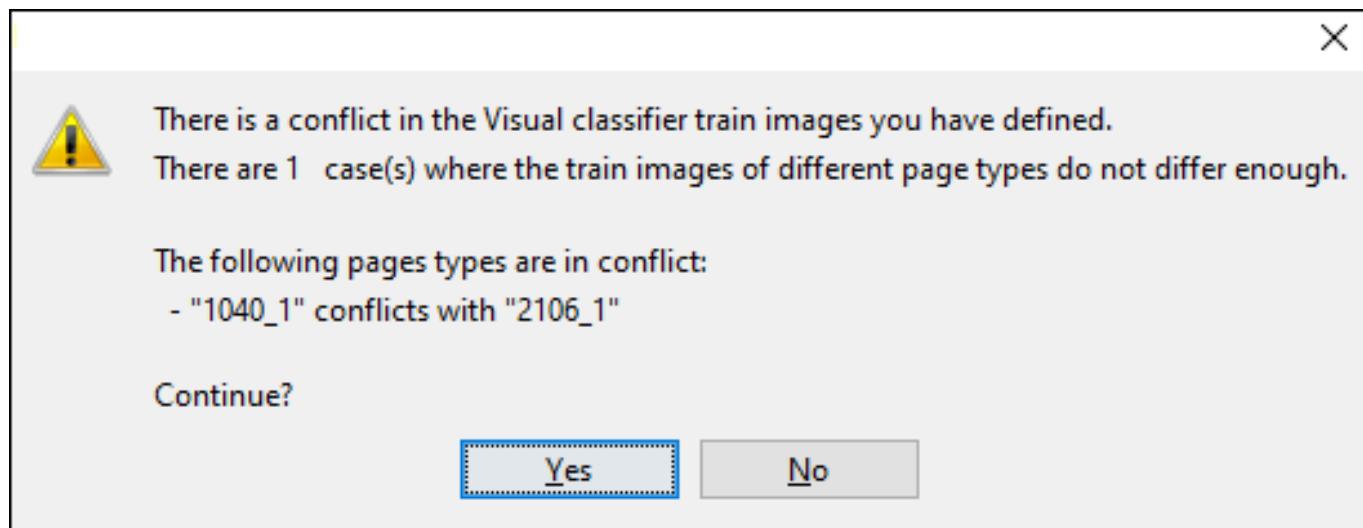
Results:			1040_1_r0107_00.tif	
File name	Type	Confidence	Class	Visual
1040_1_r0107_00.tif	[FAIL]		1040_1	90.83
1040_1_r0108_00.tif	[FAIL]		2106_1	84.59
1040_1_r0109_00.tif	[FAIL]		1040_2	56.94
1040_1_r0110_00.tif	1040_1	93.15	2106_2	50.36
1040_2_r0141_01.tif	1040_2	93.25	2441	48.86

In this case for example, the 1040_1 Page Type with a confidence score of 90.83 is only ~6% higher than the second candidate Page Type 2106_1 which is less than the configured 10% difference from next best. If however there are multiple candidate Page Types but the top candidate's confidence score difference from the second candidate is greater than 10%, then classification is successful despite having more than one candidate Page Type.

Results:			1040_1_r0110_00.tif	
File name	Type	Confidence	Class	Visual
1040_1_r0107_00.tif	[FAIL]		1040_1	93.15
1040_1_r0108_00.tif	[FAIL]		2106_1	81.09
1040_1_r0109_00.tif	[FAIL]		1040_2	56.51
1040_1_r0110_00.tif	1040_1	93.15	2441	52.88

The Classification Designer attempts to preemptively detect such cases and will produce a warning,

informing the user of which Page Types have conflicts with each other.



It is usually advised to alleviate such mistakes, except if explicit knowledge of the situation should permit this.

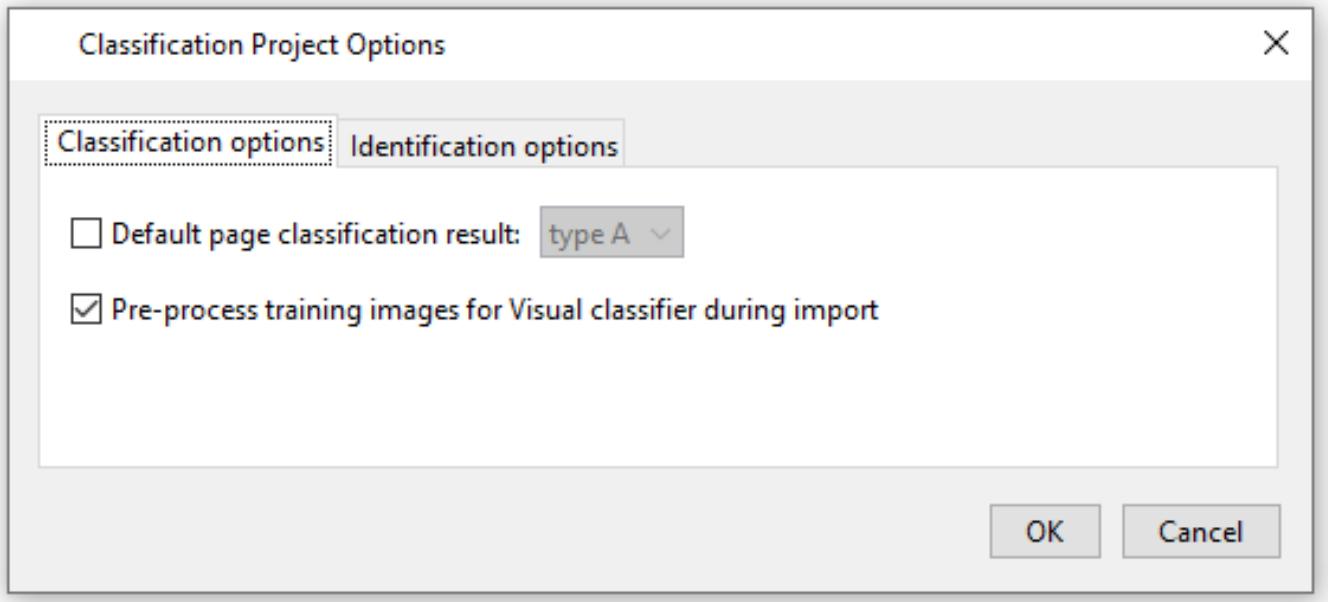
Although during testing it is important to assert the results for all classifiers, at runtime, a successful classification from the Visual classifier will not trigger the Feature classifier when defined, since the classification result has already been generated. Unchecking the bottom left checkbox, *Execute exhaustively all classifiers*, will emulate this runtime behavior.

When iterating through different test sets, saving them for later use will increase the test cycle speed. Once a test set of images is loaded, it can be saved as a quickly accessible set via the *Sets* button. Loading, updating or deleting a set is also available.

If the test results are not optimal, changes will need to be done to the *Classification project* configuration. After making any new change and while keeping the *Test* dialog open, the *Retrain* button will be available, which will quickly trigger the training of the Classification project.

Classification Project options

Apart from the Page Type and classifier specific configurations available, the Classification project offers some generic configuration options regarding classification and identification. The options panel can be accessed via the *Options* button at the top left.



Classification options

Two options are available under Classification options:

1. The *Default page classification result* option, if enabled, will set the selected Page Type (named as per its type alias) as the Page classification result in case the classifiers cannot successfully determine its Page Type. This should optionally be used as a fallback option, if it applies to the use case. Disabled by default.
2. *Pre-process training images for Visual classifier* during import dictates if the imported training images will go through an image cleanup process when they are added to the training set of the Visual classifier. The most important part of the cleanup process is the fact that the image is binarized (converted to black & white) right when it is imported. It is important to be mindful of this, especially for images that have shaded backgrounds or large color patches. Always check the result of the imported binarized images to make sure that no loss of important information has occurred. If the pre-process flag is turned off then the images are not binarized but are instead converted to grayscale (applies for color images). This option is enabled by default.

Identification options

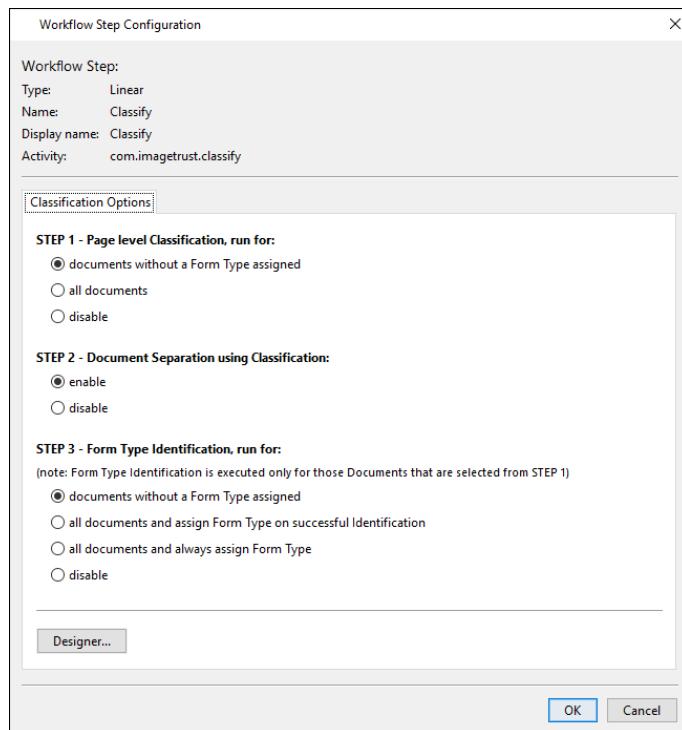
Two options are available under Identification options:

1. Similarly to the *Default page classification* option, the Default Form Type identification result will set the default Document Form Type if its identification fails. The list of available options depends on the defined Form Type names. Disabled by default.
2. If the *ignore documents without page types during identification* option is enabled, the identification step will ignore all classification Form Types that don't have any Page Types defined. This is impor-

tant because an empty classification Form Type could optionally match any Document (since its Pages do not matter) or none. The flag allows control over this behavior. Enabled by default.

Finalizing the Classify step

Once the Classification project is properly configured and saved (using the *Save* button), the final step for the Classify Workflow step setup is to configure the available Job Workflow step options. The *Gear* icon on the Classify Workflow step will open the following dialog.



The Classification Options are divided into three categories, based on the three steps followed by the classification process.

Step 1

This step defines which Batch Documents should go through the Page Type detection process. Depending on the order of the workflow steps, some Documents might already have their Form Type assigned to them before going through classification. It is possible to filter out those Documents, since manual identification has already been performed, speeding up the classification process. Note that in such a case the Document Pages will not have their Page Types assigned to them thus lacking the mapping with the Form Type Sample Pages. This means that automatic extraction will occur only for the Sample Pages that are not mapped via a classification type alias and instead have their absolute position defined.

Step 2

Mostly self explanatory, allows to optionally disable the document separation process after classification.

Step 3

The final step, Form Type identification, can be disabled, executed on all Documents or only on those that did not have a Form Type already assigned to them before reaching the Classify step (as in Step 1). If all Documents are selected for identification then a new Form Type can optionally be assigned to a Document only after a successful identification, keeping its pre-existing Form Type (which might be empty) otherwise. If a Form Type is to always be assigned after the identification step, then it is possible that a Document with a manually assigned Form Type will end up with no Form Type in case of identification failure.

Understanding Document separation

Although it is possible to separate a scanned stack of paper into their respective Documents during the Scan step (e.g. through barcode separation), this is not mandatory if a Classify workflow step is configured. When the Batch reaches the Classify step, the provided input is each Document, one by one. If no separation has occurred during the initial scanning, then only one Document will be available and will be forwarded directly for classification of its Pages. Once page classification is over, and if enabled in the workflow step configuration, the separation algorithm is as follows:

1. Find all Pages that were classified as First position Page Types. A first separation occurs at those points.
2. If a First Page is at some point followed by a Last Page (not needed to be adjacent) then separation occurs on the Last Page.
3. If there is no Last Page but the Form Type has a maximum Page Count defined then separation occurs at the Page with index matching the maximum page count.

Following the algorithm properly, it is clear that separation might produce intermediate Documents that contain Pages that either have not been classified successfully or those that were, did not belong in a Document that started with a First Page. Note that this doesn't mean that those Documents will not go through the identification process. If an intermediate Document is generated and it ends up matching with a Form Type, according to the rules explained below, then it will be properly identified as such.

Understanding Form Type identification

The final operation of the Classify workflow step is the Form Type identification attempt for each Document. The identification process is based on the positions assigned to the Page Types. If a subset of the Pages of a Document cover the entirety of the configured Page Types of a specific Form Type and additionally, if those Pages are positioned in a way that matches the Page Type positions for that specific Form Type then the Document is successfully identified. In the case where a Form Type has its mini-

um or maximum page count defined then this criterion is also taken into consideration deciding on the potential Form Type.

Examples:

- Assuming the 2106 Form Type:



Any Document of any length that begins with a 2106_1 Page and ends with a 2106_2 will be successfully identified as a 2106 Form Type.

- Assuming the 1040 Form Type:



Any Document of any length that at some point contains a 1040_1 Page and ends with the 1040_2 Page will be successfully identified as a 1040 Form Type.

- Assuming the 2441 Form Type:



Any Document of length between 1 and 5 Pages, one of which is the 2441 Page and it is neither in the 1st nor 5th position will be successfully identified as a 2441 Form Type.

Understanding the relationship between the Form Type Sample Pages and Classification project Page Types

When the Batch moves to the Extraction Workflow step, each Document Page that has index fields

must be mapped to its respective Sample Page of the Document Form Type. This is done in order to decide which zones will be extracted on that specific Page, the zones being defined in the Form Type Sample Pages. If the Sample Pages have strict positions defined, then mapping is done via Document Page positions (e.g. the 3rd Page in the Document will be matched with the Sample Page that has 3 as an absolute position property). If however there are no strict rules relating to where the Pages will be in the Document, then the only way to map them is to understand (via image classification) what Page Type a Page represents and then look via an explicitly defined mapping which of the Sample Pages represents that Page type in the Form Type. This explicit mapping is done via the classification alias string Sample Page property.

If no mapping existed (either via absolute positions or via type alias) then during extraction it would be impossible to know which zones to extract from which Document Pages resulting in random and probably wrong extraction results.

Maintaining the Classification project

It is very important to understand the need for proper maintenance of the Classification project. Although its initial setup might work fine, it is very likely that during its lifetime images will be processed by the system that fail to classify when they should have. Those cases are prime candidates for enhancing both the training and test image sets. Images that fail should be gathered and introduced to the Classification project either as training images in the Visual classifier, or the Feature classifier should be re-evaluated in order to achieve successful classification.

Along with each periodic maintenance, the project should go through a test cycle in order to guarantee not only that its classification rate has not declined but to guarantee that its performance has not waned either. The performance metrics at the bottom right of the test dialog (Performance: avg=157ms, min=87ms, max=347ms) should be noted down after every upgrade of the Classification project and should be compared with previous results in order to guarantee optimal performance.

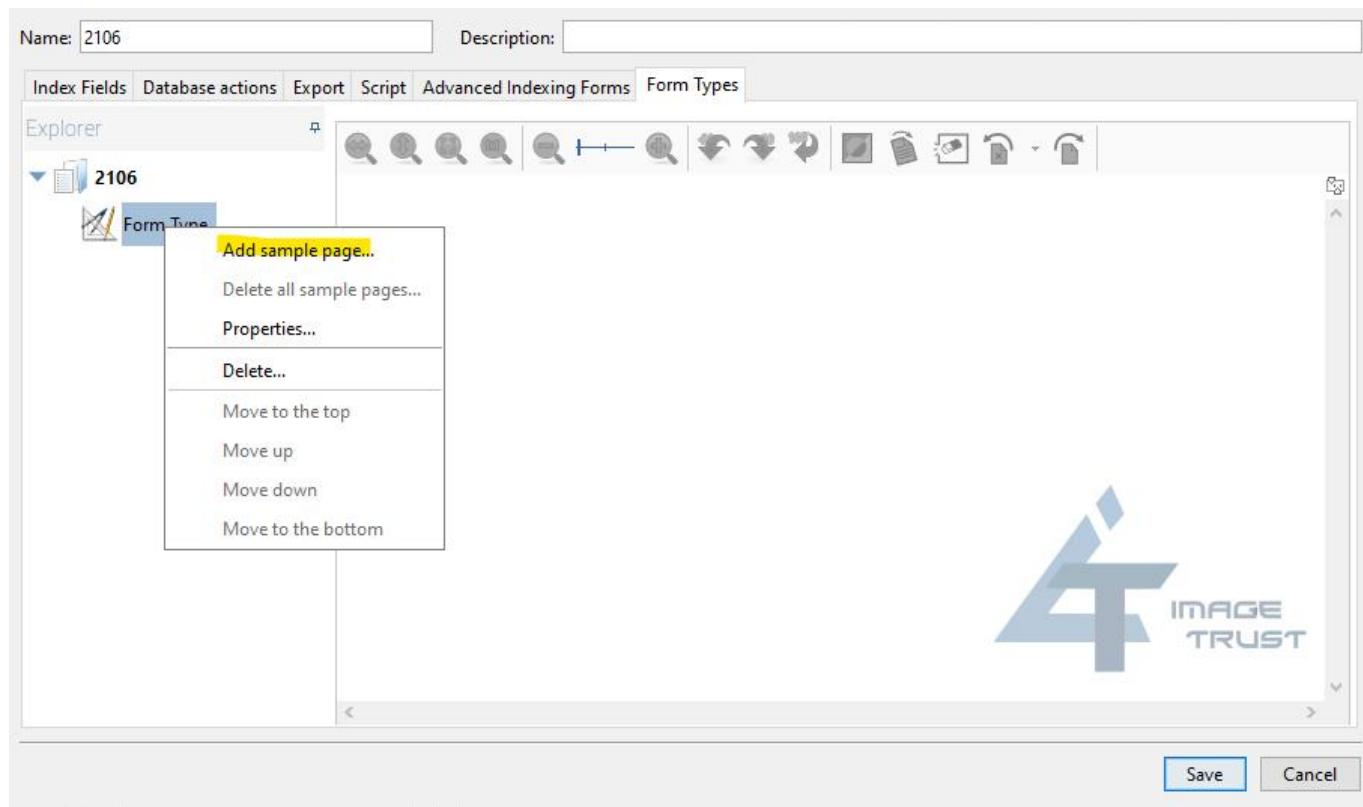
Configuring the Extract step

After the successful creation of the Classification project, it is time to move forward to the creation and configuration of the Extract Step. For the sake of this example the Previously created document classes will be used. Each will be configured with two Index Fields of *String* type named "Field1" and "Field2".

The screenshot shows the 'Indexing & Export' tab selected in the top navigation bar. Below it, a section titled 'Batch level indexing & export' contains a checkbox for assigning default values to empty index fields. A note explains that this option assigns a default value to every empty index field on all nodes of the batch when indexing starts. Two sections for 'folder' and 'document' classes are shown, each with a list of available classes: 1040, 2106, and 2441.

Below this, a table configuration screen is displayed. It includes fields for 'Name' (1040) and 'Description'. The 'Index Fields' tab is selected in the top navigation of the table view. The table itself has columns for Name, Display Label, Help Tip, Field Type, Default Value, RegEx Valid..., Required, Hidden, Read-only, and Sticky. Two rows are present: 'Field1' (String type) and 'Field2' (String type). At the bottom of the table view are buttons for 'Field Types...', 'Insert Table...', 'Edit Table...', 'Insert', 'Duplicate', 'Delete', and 'Clear'. To the right of the table is a vertical toolbar with up and down arrow buttons. At the very bottom are 'Save' and 'Cancel' buttons.

The Nuance OCR engine will be used for the Extraction. After Navigating to one of the available *Document Classes* and then navigating to the form type tab, we can add a sample page to our Form.



By right clicking the Sample Page that was previously added, we can create Index Zones. Double clicking the created Index Zone brings us to the *Index Zone Properties*. Here, an *Index Field* can be assigned to the Index Zone as well as the OCR engine that will be used.

The dialog is divided into several sections:

- Associated Index Field:** A dropdown menu set to "Field1".
- Name:** A text input field set to "Field1".
- Color:** A color selection button with a blue square preview.
- Extraction method:**
 - Use Extraction Profile:** A radio button selected, with a dropdown menu showing "OCR: Nuance OCR" and a "Edit Extraction Profiles..." button.
 - Perform image cleanup by masking out sample page snippet:** An unchecked checkbox.
 - Use result from zone:** A radio button with a dropdown menu set to "<no zone>".
- Minimum confidence:** A numeric input field set to 75 with up/down arrows.
- OCR settings:**
 - Use fuzzy regular expression on the extracted value:** An unchecked checkbox.
 - Match pattern:** An empty text input field.
 - Substitution pattern:** An empty text input field.
 - Case sensitive:** A checked checkbox.
 - Minimum regular expression similarity:** A numeric input field set to 75 with up/down arrows.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

To summarize

The previously created Classify→ Extract workflow will do the following:

- Classify documents to a Document class based on our classification project.
- After the Document Class has been assigned the two index fields of the assigned document class will be populated from the Extracted Data of our Index Zones.

3.1.11.6. Intelligent OCR Engines

Intelligent OCR Engine fundamentals

A server-side Intelligent OCR Job Workflow is an automated document recognition and extraction module that uses Machine Learning technology in order to identify specific document types and structures, such as invoices and forms, or perform full text OCR in documents. Compared to the related Classification and Extraction modules of the application server, when setting up an Intelligent OCR project, there is no need to add a sample Form Type document, design index zones and train the engine with a set of training images.

In v6.1 we have added support for the following Intelligent OCR cloud services:

- Google Document AI (<https://cloud.google.com/document-ai>)
- Microsoft Azure Form Recognizer (<https://azure.microsoft.com/en-us/services/form-recognizer/>)

Benefits from using a Job Workflow with an Intelligent OCR step

- Design your Workflow faster and with less effort.
- Quickly extract text, key-value pairs and structure from documents.
- Ensure your data is accurate and compliant.
- Automate and validate your data to make your workflows more efficient.
- Automate data capture at scale to reduce document processing costs.
- Make better and faster decisions using document data.
- Rely on Google and Microsoft's security models and world-scale infrastructure to keep your organization secure.
- Improve operational efficiency by extracting structured data from unstructured documents and making that available to your business.

Setting up a project

Setting up an automatic Intelligent OCR project requires several configuration steps. An Intelligent OCR Workflow step can be combined with all different Workflow step types provided by the application, even with Classification and Extraction steps. Two common use cases of Intelligent OCR are the following:

- Identify and extract values from structured documents such as invoices and forms.
- Perform full-document OCR extraction (or for a specific number of leading pages in the document) and use the extracted data at a following step, e.g with an Images Export configuration step.

Step 1: Understand the content that will be handled

The first step should be understanding what content will be handled by the Job, and specifically from the Intelligent OCR Workflow Step and gathering the requirements for the project. That includes:

Getting a list of all types of documents that will be handled by the Job. A different extraction profile needs to be created for every different type of document that is going to be processed by the Intelligent OCR engines.

Google Document AI engine offers three types of document processing:

- Form parser
- Invoice parser
- Document OCR

Microsoft Azure Form Recognizer offers five types of document processing:

- Invoices
- Sales receipts
- Business cards
- Identification cards
- Document OCR

Step 2: Identify and configure the required index fields in Job Setup, at Document Class level



This step can be skipped in the case of full Document OCR Extraction, for searchable PDF Export. Otherwise, all the returned values, for example from an invoice, must be represented in Document Class configuration as index fields.

1. Identify the type of data that each field contains and create the appropriate Field Types (e.g. String, Date, Integer, List, etc).
2. For each field, define and create the respective Index Field. For further information about Index Field setup, refer to [Indexing](#) section.

Step 3: Setup an Intelligent OCR Workflow step

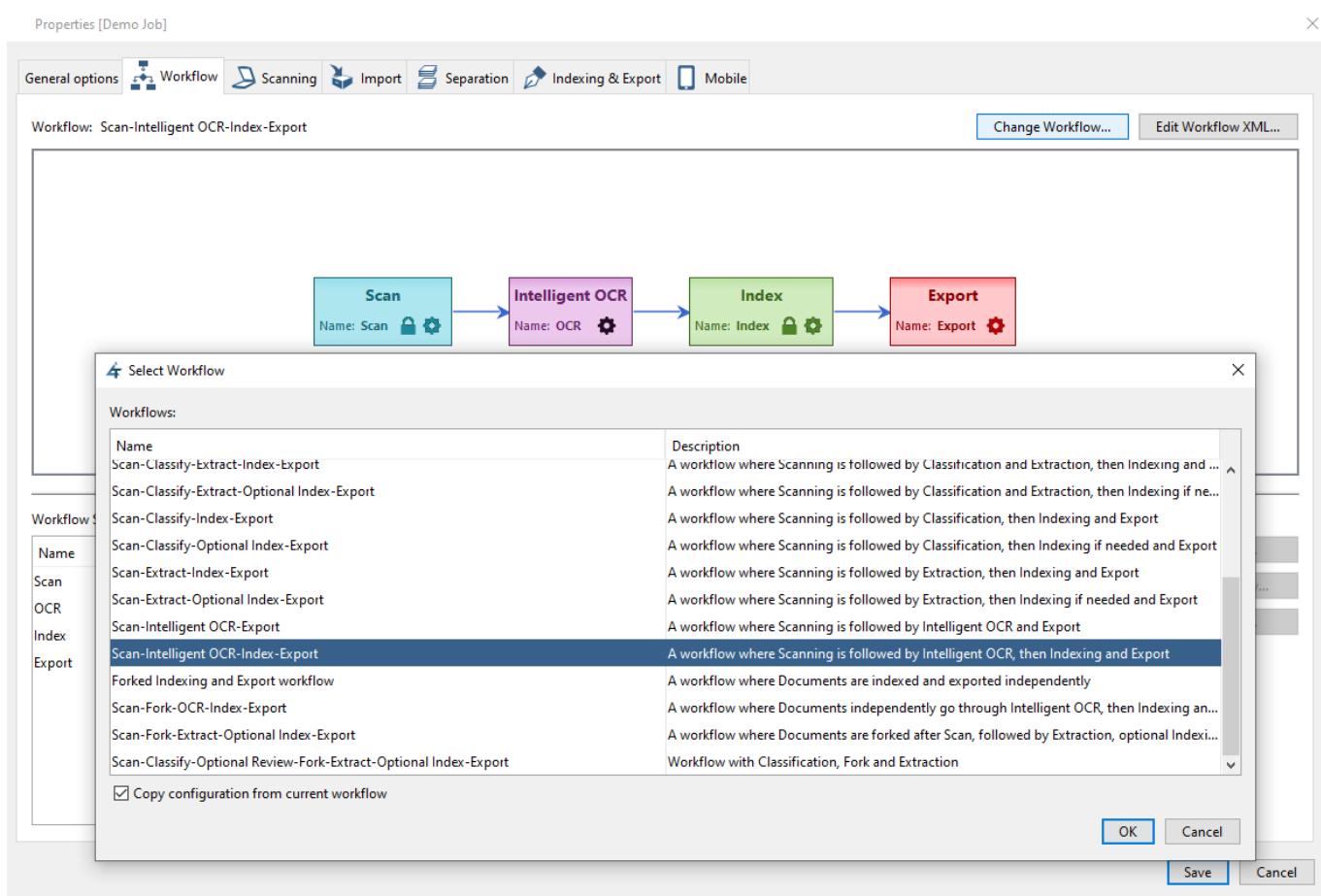
The Intelligent OCR Workflow step is available during Job configuration.



Figure 50. Intelligent OCR Workflow step

3.1.11.7. Intelligent OCR Workflow step

The quickest way to setup a workflow containing Intelligent OCR is by selecting one of the present Workflows by pressing the Change Workflow button. Of course, the Workflow can also be edited, to be combined with other steps.

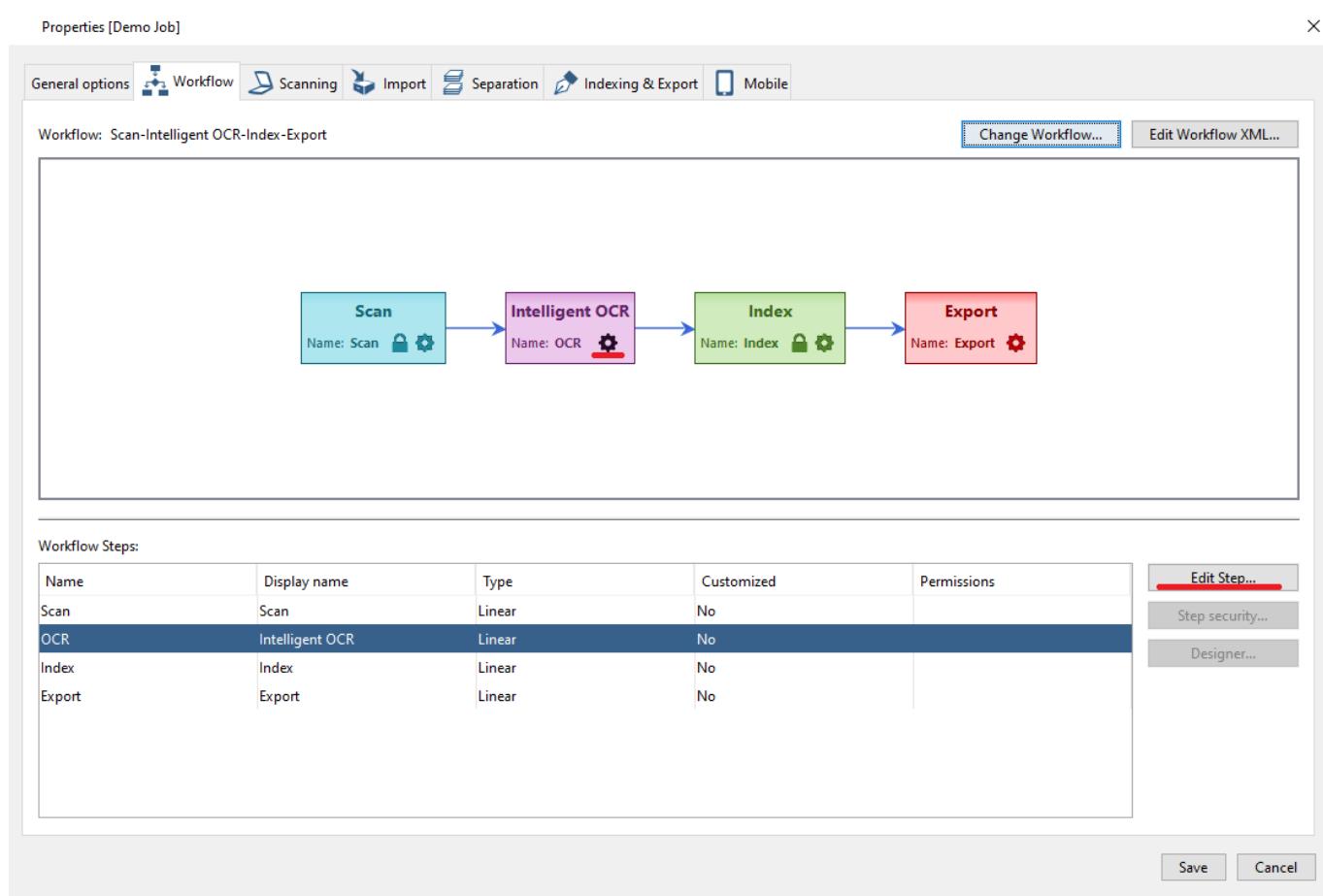


The steps are set up and ready to be configured.

Creating an Intelligent OCR Project

General configuration

The configuration of the Intelligent OCR step is the most important part of a successful automatic recognition and extraction workflow. A correct Intelligent OCR process means less manual Indexing. To achieve this, you need to clarify the exact type of document that is going to be processed and whether it will be a key-value extraction or a full text OCR. According to those two conditions, an extraction profile that fits the business requirement needs to be created. To open the Intelligent OCR configuration window, click on Edit Step button or the gear icon on the Workflow grid.



Workflow Step Configuration X

Workflow Step:

Type: Linear
Name: OCR
Display name: Intelligent OCR
Activity: com.imagetrust.ocr

Extraction Profile selection Script Mappings Options

Default mode

Extraction Profile: Intelligent OCR: Google AI Invoice Edit Extraction Profiles...

Apply to: All Documents
 Documents with **any** Form Type
 Documents **without** Form Type

Advanced mode

Selector	Extraction Profile
[All Documents]	[None]
[Any Form Type]	[None]
[No Form Type]	[None]

Note: more specific selectors have higher priority, starting with [specific <Form Type>], then [specific <Document Class>], then [Any Form Type], then [No Form Type], then [All Documents]. Unless there is an entry for [All Documents] it is possible that some Documents will not be processed, if they don't match any other selector.

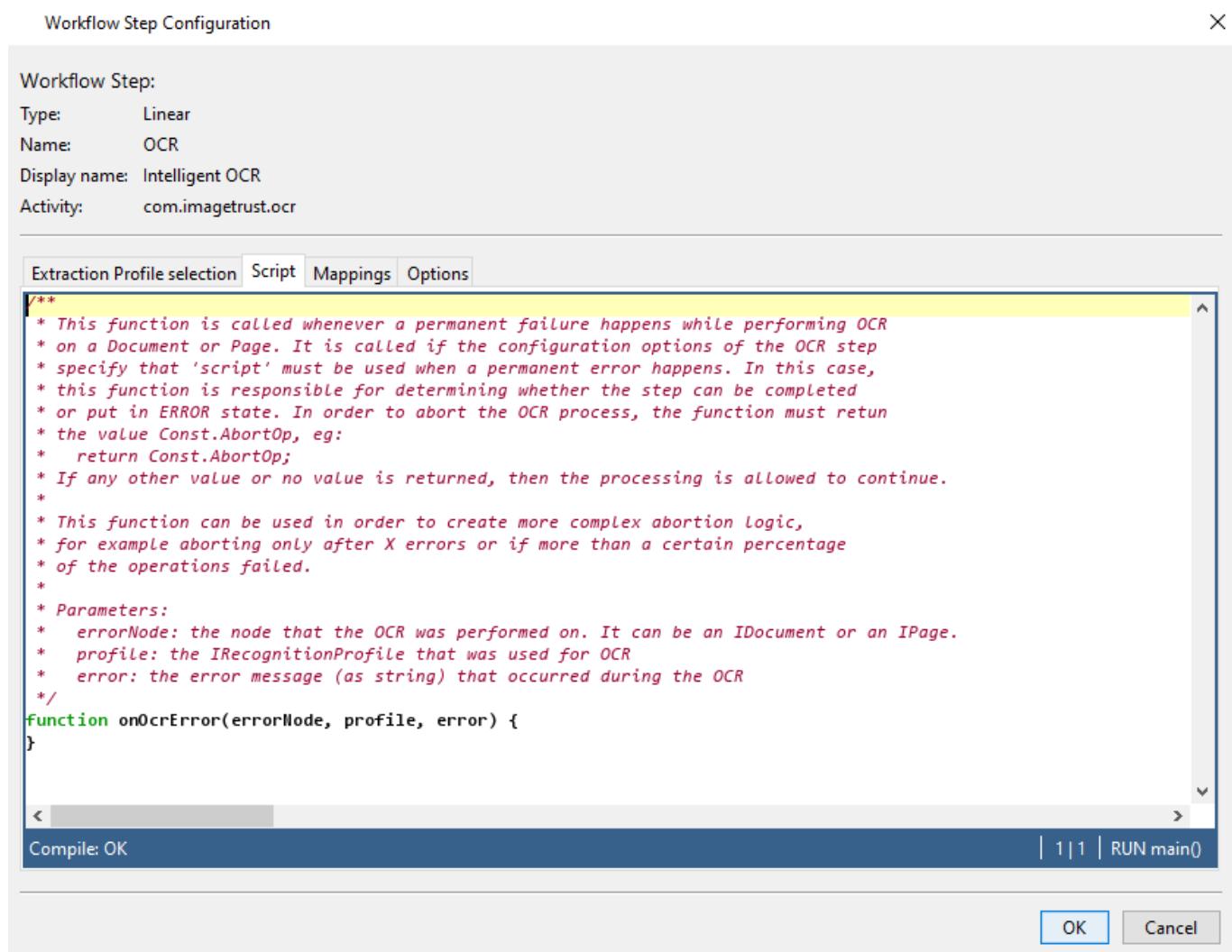
OK Cancel

1. Default mode The same extraction profile will be used for all processed documents. The selected extraction profile must support the action that it is meant to perform. For example, for invoice recognition and extraction, an extraction profile with Invoice parser (Google Document AI) or Invoices (Microsoft Azure Form Recognizer) recognition mode must be selected. In case of full document OCR, an Extraction Profile with one of the traditional OCR engine options (Microsoft Computer Vision, Google Vision, Amazon Textract, Nuance OCR) can also be used for this iOCR Workflow Step configuration. The "Apply To" option decides on which of the documents the Intelligent OCR will run. By default, it will run for all documents. In a Job Setup with multiple Form Types (and/or Document Classes), you can select the Documents that OCR will run for, according to whether they have a Form Type assigned or not. The Form Type assignment might have been made by a Classification Step that precedes the Intelligent OCR step.
2. Advanced mode A different extraction profile can be selected for Documents with a Form Type, without Form Type or all Documents. Note: more specific selectors have higher priority, starting with [specific <Form Type>], then [specific <Document Class>], then [Any Form Type], then [No Form Type], then [All Documents]. Unless there is an entry for [All Documents] it is possible that some

Documents will not be processed, if they don't match any other selector.

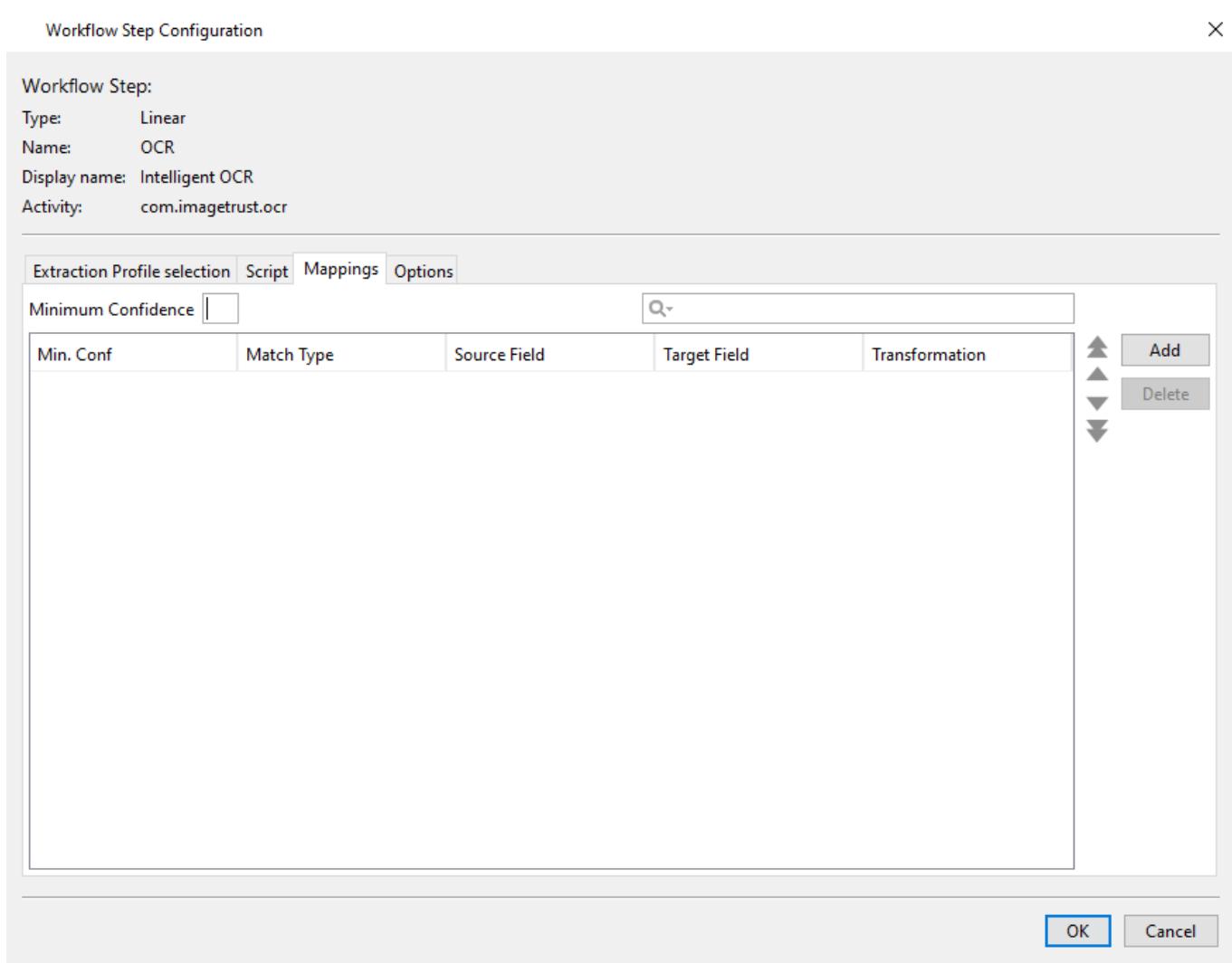
Script:

In the second tab, a custom scripting logic can be added for cases where the default process fails for some reason. The function is called whenever a permanent failure happens while performing OCR on a Document or Page. It is called if the configuration options of the OCR step specify that 'script' must be used when a permanent error happens. In this case, this function is responsible for determining whether the step can be completed or put in ERROR state.



Mappings

In the third, and most critical tab, the mapping between the extracted results and the *Info Input Solution* index fields will be configured along with custom settings.



1. Minimum Confidence: The minimum accepted confidence for the values returned from the OCR engines. In case the returned value's confidence is lower than the defined one, the index field will be flagged as Invalid (Review required). The confidence value can be pre-defined for all different mappings in total, or it can be edited row by row.
2. Match Type: The type of match between the source Field and the Target field. The available options are:
 - Exact (literal use of text)
 - Regex (using Regular Expressions)
 - Advanced (using scripting logic).
3. Source Field: The field name or label or property that comes as OCR result from the Intelligent OCR engines and can be used as standard identifiers of a key-value pair.
4. Target Field: The index field in Info Input Solution that will be populated with the Source Field value.
5. Transformation: This column can be used to add some custom scripting logic to transform the received value to a different value or calculate some values and produce a new result.

Options

General options regarding the Intelligent OCR functionality. These settings will vary from one use case to another.

Workflow Step Configuration X

Workflow Step:

Type: Linear
Name: OCR
Display name: Intelligent OCR
Activity: Intelligent OCR

Extraction Profile selection **Script** **Mappings** **Options**

Only process the first Pages of each Document

If Document does not contain all OCR data from the selected Extraction Profile:

Re-process fully, overwriting existing data
 Do not process again and use partial data to run field mappings and/or scripts

If Document contains all OCR data from the selected Extraction Profile:

Do not process again
 Re-process fully, overwriting existing data
 Use existing data and only process field mappings and/or scripts

Page image selection:

Default image
 Image with alias: Var

If image with alias does not exist: Use default image
 Stop processing (error)

When an error on a Document happens:

Apply following retry logic before assuming this is a permanent error:

Maximum retry attempts:

Wait between attempts: seconds

If the error is permanent, then:

Abort
 Continue
 Use script

OK **Cancel**

1. Only process the first _ Pages of each Document: This checkbox, if enabled, can decide how many pages of a Document will be processed during Intelligent OCR. This will reduce the cost of processing a large document, if not all pages need to be extracted. Note that, in case of full Document OCR

where the extracted data will be used for searchable PDF generation, the same number of pages selected here, will be searchable at the exported PDF. For example, if the number in this selection is 2, the exported PDF will only have its first 2 pages as searchable (text layer).

2. If Document does not contain all OCR data from the selected Extraction Profile: This option dictates what happens if a Document contains no (or partial) OCR data from a previous execution of the same Extraction profile and it is called to be processed again. The Document might not contain the full OCR data due to a change to the Batch structure. For example, the documents of a batch that was OCRed in an Intelligent OCR step and then, were split into separate documents in a Classification step, will not contain the full OCR data extracted in the first step. In that case, the available options are:
 - a. Re-process fully, overwriting existing data: The document will lose all previous OCR data and the process will start again. That will cause additional calls to the OCR engine's provider.
 - b. Do not process again and use partial data to run field mappings and/or scripts: The document will use the partial (or no) OCR data that have been extracted in the previous execution. This choice will not cause any additional calls to the OCR engine's provider but the OCR data may not be full.
3. If Document contains **ALL** OCR data from the selected Extraction Profile: This option dictates what happens if a Document contains the full OCR data from a previous execution of the same Extraction profile, and it is called to be processed again. This could happen either because of an error during OCR or in a use case with more than one Intelligent OCR steps. The available options are:
 - a. Do not process again: The document will retain the extracted data from the previous process. This will not cause additional calls to the OCR engine's provider.
 - b. Re-process fully, overwriting existing data: The document will lose all previous data and the process will start again. This will cause additional calls to the OCR engine's provider.
 - c. Use existing data and only process field mappings and/or scripts: The Document will keep the full OCR data extracted from the previous process and the extraction will run again for the mappings and/or scripts. If for any reason some OCR data are missing, the Document will be re-processed fully, and it will cause additional calls to the OCR engine's provider.
4. When an error on a Document happens: This option refers to the error handling logic that will be followed. The first option is how many times a task will be retried before it is considered as a permanent failure and the second one is the time that will pass between one re-try and another.
5. If the error is permanent, then: This option decides what happens when a failure is permanent. The available options are:
 - a. Abort: The task will be moved to error state.
 - b. Continue: The task will move to next Workflow step even if the process failed.
 - c. Use script: The scripting logic of the second configuration tab will be followed.

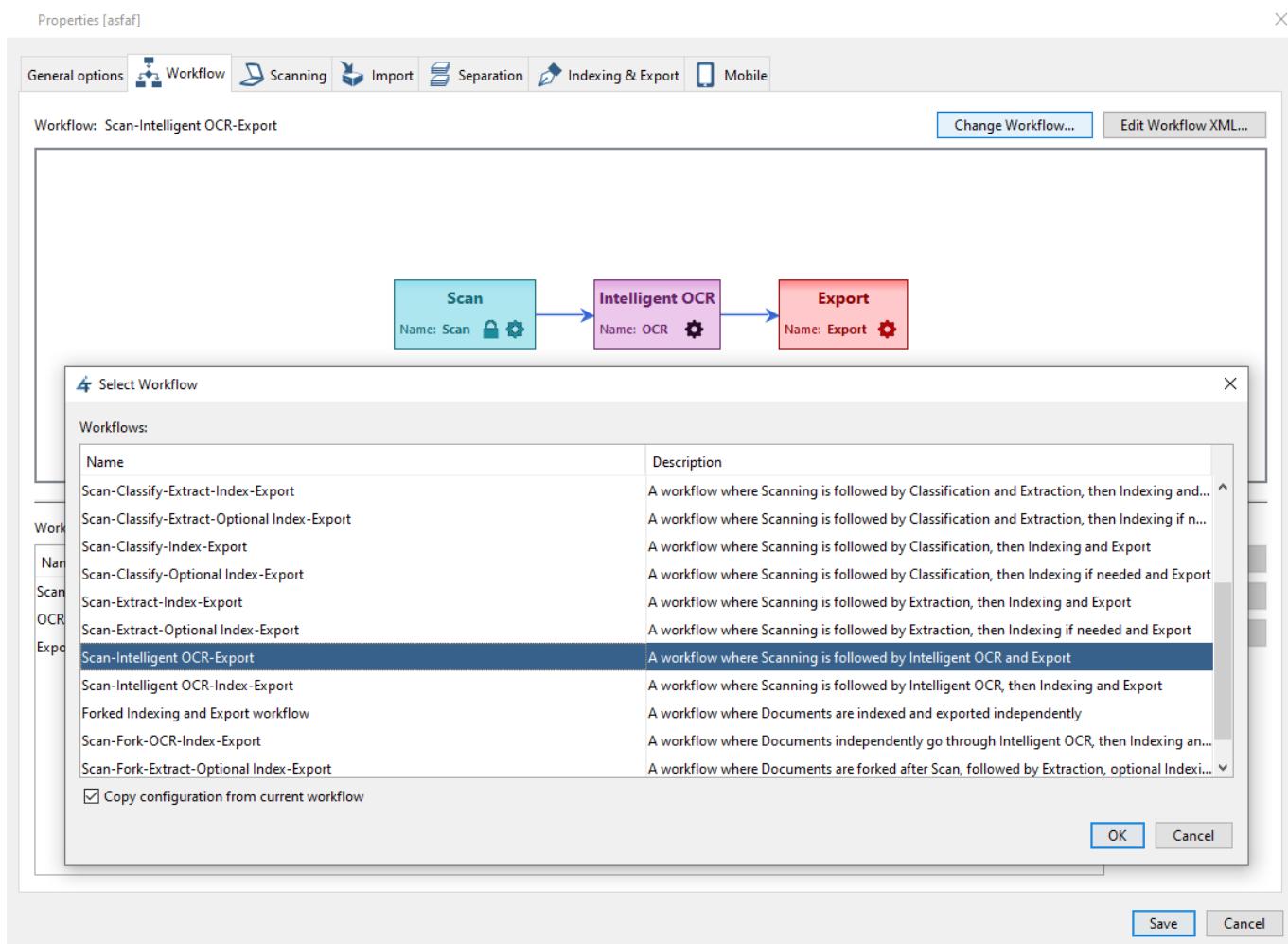
In the following sections, two different business examples will be demonstrated. The configuration is similar for all other use cases.

Example 1: Full document OCR

This implementation will perform a full-text document OCR in unstructured documents. The main purpose of such an implementation is to use the OCR results in a later Workflow step. For example, an Images Export configuration at Export step would be able to use the produced data and create a searchable PDF. In this implementation, it is not mandatory to use one of the two Intelligent OCR engines (Google AI, Microsoft Form Recognizer). An Extraction Profile with one of the traditional OCR engine options (Microsoft Computer Vision, Google Vision, Amazon Textract, Nuance OCR) can also be used.

Step 1: Create a Job Workflow:

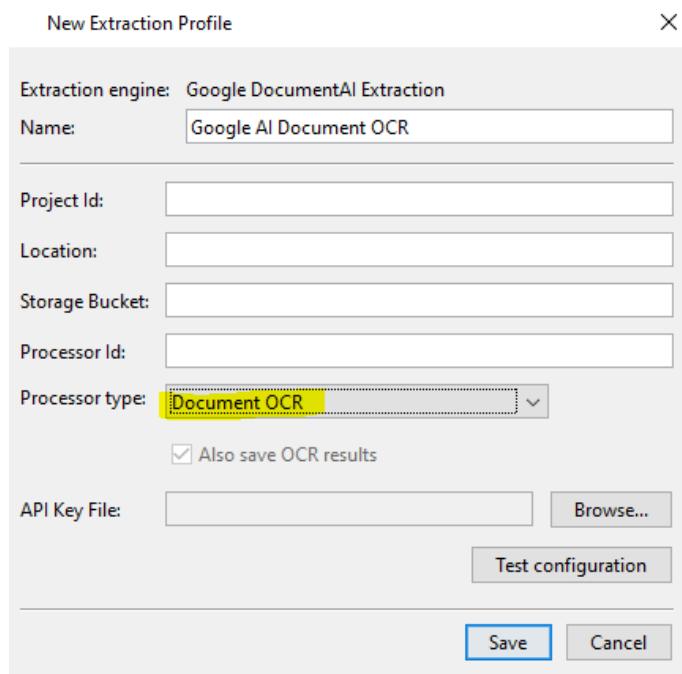
- From *Info Input Solution Tools & Options* menu at the top right corner, click on Job Setup.
- Click on the plus icon to create a new Job Setup, name it and click Save.
- Move to Workflow tab and click on "Change Workflow" button. Select the Scan-Intelligent OCR-Export Job Workflow and click OK, as shown below:

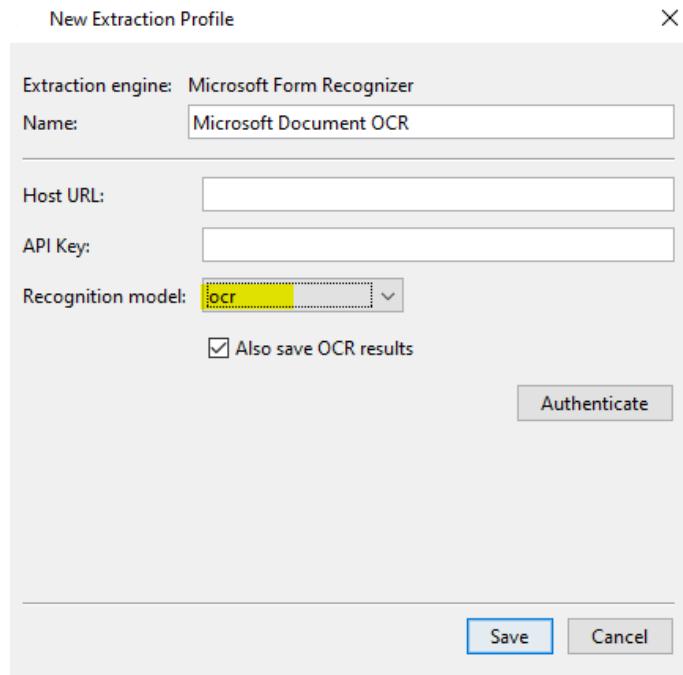


- Click on Edit Step button or the gear icon on the Workflow grid to open the Intelligent OCR configuration window.

Step 2: Create an extraction profile:

- After opening the Intelligent OCR configuration window, move to "Extraction Profile Selection" tab and click on "Edit Extraction Profiles..." button.
- Click on the plus sign to create a new extraction profile.
- From the OCR or Intelligent OCR sections, select the engine to be used. In case of Intelligent OCR, the following Processor Type or Recognition model must be selected, for Google AI and Microsoft Form Recognizer engines accordingly.



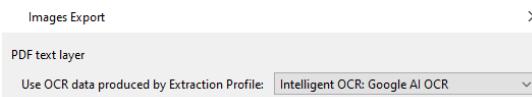


Step 3: Select extraction profile:

Back to the Intelligent OCR configuration window, at "Extraction Profile Selection" tab, select the extraction profile created in Step 2. Configure the rest of the settings as explained above, save the configuration and close the window.

Step 4: Create a searchable PDF export configuration:

- From Job Setup properties window, move to Indexing & Export tab and click on "Batch level Indexing & export" button.
- Move to Export tab and click on "New" button.
- Select "Images Export" and in the opened window select PDF as file format.
- Then, click on "PDF Options" button.
- On the "PDF text layer" section, select the extraction profile referenced in Step 1 and Step 2.



- Complete the rest of the Images Export properties, save the changes and publish the Job Setup.

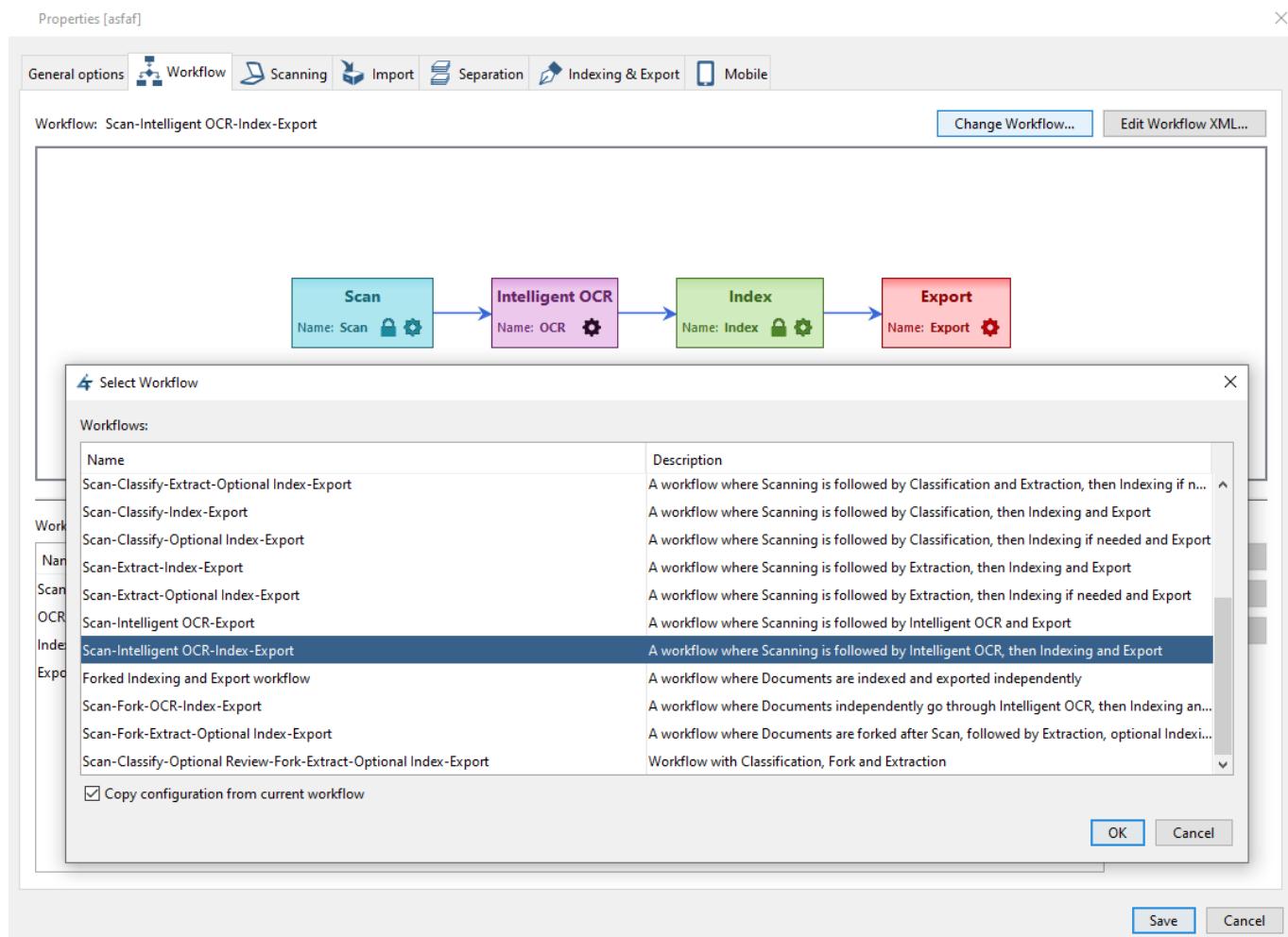
The Job Workflow is ready to process any document using full Document OCR and produce a searchable PDF at batch level Images Export.

Example 2: Invoice recognition and extraction

This implementation will perform identification and extraction for any given invoice. Then, it will assign the extracted values to the corresponding *Info Input Solution* index fields. It is assumed that a Document Class and the appropriate index fields are already created following the instructions at [Indexing](#) section.

Step 1: Create a Job Workflow

- From Info Input Solution Tools & Options menu at the top right corner, click on Job Setup.
- Click on the plus icon to create a new Job Setup, name it and Save.
- Move to Workflow tab and click on "Change Workflow" tab. Select the Scan-Intelligent OCR-Index-Export Job Workflow and click OK, as shown below:



- Click on Edit Step button or the gear icon on the Workflow grid to open the Intelligent OCR configuration window.

Step 2: Configure extraction profile

- After opening the Intelligent OCR configuration window, move to "Extraction Profile Selection" tab and click on "Edit Extraction Profiles..." button.
- Click on the plus sign to create a new extraction profile.
- From the Intelligent OCR section, select the desired engine. The following Processor Type or Recognition model must be selected, for Google AI and Microsoft Form Recognizer engines accordingly.

New Extraction Profile X

Extraction engine: Google DocumentAI Extraction

Name:

Project Id:

Location:

Storage Bucket:

Processor Id:

Processor type: ▼

Also save OCR results

API Key File: Browse...

Test configuration

Save Cancel

New Extraction Profile X

Extraction engine: Microsoft Form Recognizer

Name:

Host URL:

API Key:

Recognition model:

Also save OCR results

Step 3: Select extraction profile

Back to the Intelligent OCR configuration window, at "Extraction Profile Selection" tab, select the extraction profile created in Step 2.

Step 4: Configure mapping

After selecting the appropriate extraction profile for Invoice OCR, move to Mappings tab. In this step, the Source Fields received from the Intelligent OCR engines need to be mapped with the corresponding Info Input Solution index fields. Regarding the Invoice recognition, each Intelligent OCR engine offers some standard, pre-defined properties which cover all the available information that can be found in an invoice document. These properties are named differently between the two engines. In this example, the Microsoft Form Recognizer engine has been used.

1. From the top right corner of the window, click on Add button to add new rows (as many as the index fields to be used).
2. Double click inside the Source Field text area to expand the dropdown list. Expanding the list with the down arrow of the keyboard will display all the available options provided by the engine.
3. Select a value from the list (e.g. InvoiceId) and hit enter.
4. Double click on Target Field text area to expand the dropdown list. All the available Info Input Solution index fields will be displayed. Select one index field and hit enter.

Repeat the above steps to map all the necessary index fields. After finishing this process, the Mappings window will look like this:

Extraction Profile selection Script **Mappings** Options

Minimum Confidence 0.0

Q-

Add ▲ ▼ Delete

OK Cancel

Min. Conf	Match Type	Source Field	Target Field	Transformation
0	EXACT	InvoiceId	InvoiceID	
0	EXACT	InvoiceDate	InvoiceDate	
0	EXACT	DueDate	DueDate	
0	EXACT	PurchaseOrder	PurchaseOrder	
0	EXACT	VendorName	VendorName	
0	EXACT	VendorAddress	VendorAddress	
0	EXACT	VendorAddressRecipient	VendorAddressRecipient	
0	EXACT	CustomerName	CustomerName	
0	EXACT	CustomerAddress	CustomerAddress	
0	EXACT	CustomerAddressRecipient	CustomerAddressRecipient	
0	EXACT	CustomerID	CustomerID	
0	EXACT	BillingAddress	BillingAddress	
0	EXACT	BillingAddressRecipient	BillingAddressRecipient	
0	EXACT	ShippingAddress	ShippingAddress	
0	EXACT	ShippingAddressRecipient	ShippingAddressRecipient	
0	EXACT	ServiceAddress	ServiceAddress	
0	EXACT	ServiceAddressRecipient	ServiceAddressRecipient	
0	EXACT	ServiceStartDate	ServiceStartDate	
0	EXACT	ServiceEndDate	ServiceEndDate	
0	EXACT	SubTotal	SubTotal	
0	EXACT	TotalTax	TotalTax	
0	EXACT	InvoiceTotal	InvoiceTotal	
0	EXACT	AmountDue	AmountDue	
0	EXACT	RemittanceAddress	RemittanceAddress	
0	EXACT	RemittanceAddressRecipient	RemittanceAddressRecipient	
0	EXACT	PreviousUnpaidBalance	PreviousUnpaidBalance	

Click OK, save the changes and publish the Job Setup. The Job Workflow is ready to process any provided invoice and return the extracted values. Some of the above options, of course, might not exist in all invoices, so some of the index fields can be left out of the configuration.

Alternative implementation: Form Parsing

In case the extracted document is not an Invoice, the Source Field values will not be provided by the Engines as a pre-configured list. In that case, the Source Field values must be deployed according to the keywords extracted from each Form cell or text field. To be accurate with the Source Field values, it is recommended to use the "Regex" Match Type option. A "Mapping" example of such an implementation follows:

Workflow Step Configuration X

Workflow Step:

Type: Linear
Name: OCR
Display name: Intelligent OCR
Activity: com.imagetrust.ocr

Extraction Profile selection Script **Mappings** Options

Minimum Confidence Q

Min. Conf	Match Type	Source Field	Target Field	Transformation	Add
0	REGEX	.*Surname.*	Surname		↑
0	REGEX	[A-Z]ame.*	Name		↓
0	REGEX	.*Middle name.*	MiddleName		Delete
0	REGEX	.*Date of Birth.*	DateOfBirth		↑
0	REGEX	.*Nationality.*	Nationality		↓
0	REGEX	.*Marital Status.*	MaritalStatus		↑
0	REGEX	.*Date of Expire.*	DateOfExpire		↓
0	REGEX	.*Date of Issue.*	DateOfIssue		↑
0	REGEX	.*International passport No.*	PassportNumber		↓
0	REGEX	.*Present Address.*	PresentAddress		↑
0	REGEX	.*Contact Number.*	ContactNumber		↓
0	REGEX	.*Contact E-mail.*	ContactEmail		↑
0	REGEX	.*School adress.*	SchoolAddress		↓
0	REGEX	.*School name.*	SchoolName		↑
0	REGEX	.*Date.*	Date		↓

OK Cancel

Alternative implementation 2: Form / Invoice Parsing with Tables

Most of the Cloud OCR engines now provide Table OCR and Extraction capabilities, both with their Invoice Parsers (Processors) and their Generic Form Parsers. The Job Setup configuration in such a scenario is similar to the previous example. When doing Invoice parsing, the Source Field values may be pre-defined while when doing Generic Form parsing they need to be configured by the Administrator. The only difference is that the index fields that will take part in the Table Extraction, must also belong to a table inside the Info Input Solution Job Setup. You can find instructions how to create a Table with a group of index fields [here](#).

After creating the Table in the Info Input Solution Job Setup, the Mapping table in the Intelligent OCR step will look like this:

Workflow Step Configuration

Workflow Step:

Type: Linear
Name: OCR
Display name: Intelligent OCR
Activity: com.imagetrust.ocr

Extraction Profile selection Script Mappings Options

Minimum Confidence 90.0

#	Min. Conf	Match Type	Source Field	Target Field	Transformation
1	90	EXACT	Invoice	InvoiceNumber	
2	90	EXACT	Date	InvoiceDate	
3	90	EXACT	Invoice Total:	TotalAmount	
4	90	EXACT	Item Number	ItemNumber	
5	90	EXACT	Description	Description	
6	90	EXACT	Unit	Unit	
7	90	EXACT	Ordered	Ordered	
8	90	EXACT	Shipped	Shipped	
9	90	EXACT	Back Order	BackOrder	
10	90	EXACT	Unit Price	UnitPrice	
11	90	EXACT	Extended Amount	ExtendedAmount	

Add Delete Remove All Auto-fill ...

OK Cancel

It is clear that the index fields that are part of a Table have a table sign next to their Target Field value.

Useful tools provided by the cloud OCR engines

Note that, both Intelligent OCR engines provide some very useful tools in order to test and compare OCR results (label- value pair).

For Google Document AI engine, use the following link:

<https://cloud.google.com/document-ai/docs/drag-and-drop>

For Microsoft Form Recognizer engine, use the following link: <https://fott-2-1.azurewebsites.net/>

3.1.11.8. Point & Click OCR

This functionality allows for the use of previously generated OCR data for *Point & Click OCR* while Indexing. Using this, the index user (or the end user) can easily populate index fields by simply clicking on the

desired field, and then clicking on the word or string that we wish to populate the field with.



By "previously generated OCR" data we refer to OCR data that exists on the page when it reaches the Indexing step. The OCR data could have been generated by an *Intelligent OCR step*, an *Extract step*, or could be already exist on the page when it was imported to the application.



This feature is only available on the *HTML-Client*

The screenshot shows an invoice document from 'META - LEGAL & FINANCE' with a lion logo. The invoice details are as follows:

Invoice No:	INVOICE_RECEIPT_00000135
Invoice Date:	12 March 2020
Purchase Order:	F0016

The invoice table lists the following items:

Qty	Item	Amount
2	ACP101 System Imports Hours spent importing customer records into Premier Version	\$220.00
3	ACP100 Accounting Package Annual Subscription to Standard Version of Accounts System	\$900.00
4.5	ACP100T Online Training Hours of Training in Standard Version - Interactive Demos with Q&A Sessions	\$495.00

At the bottom, the total is \$5,715.00, payment terms are 14 days, and the payment due date is Thursday, 26 March 2020.

On the right, a sidebar shows fields for data extraction:

- InvoiceNumber
- InvoiceDate
- Total
- Field4 (MM/dd/yyyy)
- Tax

Blue lines connect the extracted text in the invoice to the corresponding fields in the sidebar, indicating the mapping for data extraction.

Qty	Item	Amount
2	ACP101 System Imports Hours spent importing customer records into Premier Version	\$220.00
3	ACP100 Accounting Package Annual Subscription to Standard Version of Accounts System	\$900.00
4.5	ACP100T Online Training Hours of Training in Standard Version - Interactive Demos with Q&A Sessions	\$495.00

Total: \$5,715.00

Payment Terms: 14 days
Payment Due By: Thursday, 26 March 2020

Figure 51. After clicking on the required data the field gets populated

Example

Suppose the OCR engine has generated OCR entities along with the full OCR data (e.g. Invoice parsing). In that case, these entities will be pre-designed on the page, for an easier and more user-friendly experience. By clicking the CTRL button, these entities are highlighted boldly so the user can locate them over the page.

3.1.11.9. Image Enhancement Profiles

The Image Enhancements Profiles can be used to perform any image operation on the batch images. For more information, please refer to the corresponding section linked below: [Image Enhancement profiles](#)

3.1.11.10. Review/Rescan Functionality

Review functionality makes possible for a user to review and reject a node providing also a related reason. Then the rejected nodes will be resolved or accepted on another step in order the batch to move

on to the designed Workflow steps. To configure the *Review* functionality select a User Workflow step (with activity scan or index) and select the tab review, see the images below,

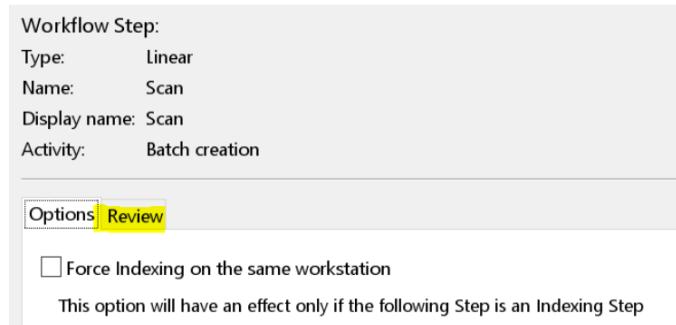


Figure 52. Review Functionality: Scan step

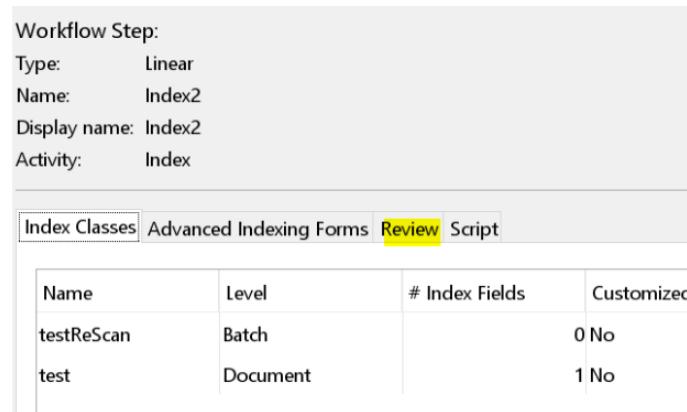


Figure 53. Review Functionality: Scan step

Here is an explanation list with all the options of the *Review* tab,

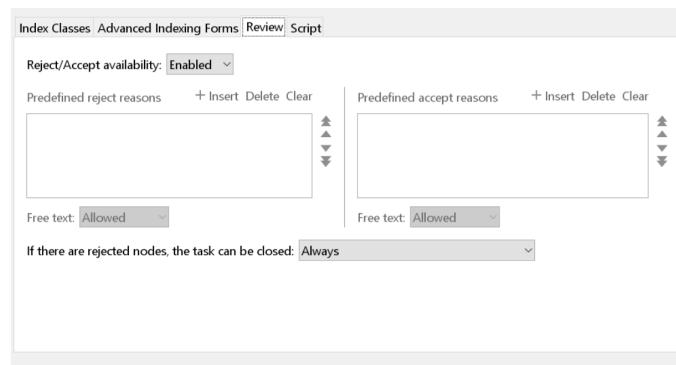


Figure 54. Review Functionality: Review tab

1. *Availability*: This option defines the main behavior of the Reject/Unreject availability for the specific step, the possible options are Enabled, Editable, and Hidden and they are described below,
2. *Predefined reject reasons* (left column): In this table it is possible to add predefined Reject reasons.

These reasons will show up to the Client GUI when Rejecting nodes. The Insert Delete and Clear buttons can be used to add, remove or delete all the records accordingly,

3. *Free text* (left Column): This option defines if the User will be able to write some comment, the possible options are Allowed, Not Allowed, and Required.
 - a. When *Allowed*, the user can decide to write a comment, it is also possible to be left blank.
 - b. When *Not Allowed*, the user cannot write any comment,
 - c. When *Required*, the user should write a comment,
4. *Predefined unreject reasons* (right column): In this table it is possible to add the predefined unreject reasons. These reasons will show up to the GUI when Unrejecting nodes. The Insert Delete and Clear buttons can be used to add, remove or delete all the records accordingly,
5. *Free text* (right column): This option defines if the User will be able to write some comment, the possible options are Allowed, Not Allowed, and Required.
 - a. When *Allowed*, the user can decide to write a comment, it is also possible to be left blank,
 - b. When *Not Allowed*, the user cannot write any comment,
 - c. When *Required*, the user should write a comment,
6. If there are rejected nodes, the task can be closed: This options can be used to define the behavior of the client for this step when there are rejected nodes. The possible values are *Always*, *Never*, and *If all rejected nodes have been updated*,
 - a. When *Always* is selected, then the task will always be able to be closed. This option can be used in steps where the user is not required to use the *Reject Functionality*.
 - b. When *Never* is selected, then the task will not be able to be closed if there are any nodes that are *Rejected*. The user will have to *Unreject* the nodes in order to close the batch,
 - c. When *If all rejected nodes have been updated* is selected, then the user can *Reject* nodes or *Unreject* nodes, it is required to update the nodes that contains a *Reject* reason before closing the batch,

In case the *Review* tab is opened for a Branching step, a new table is shown at the bottom of the window. See the image below,

Workflow Step:

Type: Branch
Name: Index
Display name: Index
Activity: Index

Index Classes Advanced Indexing Forms Review Script

Reject/Accept availability: **Editable**

Predefined reject reasons		Predefined accept reasons					
		+ Insert	Delete	Clear	+ Insert	Delete	Clear
Bad quality scanned page					Page rescanned		
Wrong page sequence					Pages rearranged		
Invalid OCRed value content					The OCRed Value is correct		
Free text: Allowed					Free text: Required		

If there are rejected nodes, the task can be closed: **If all rejected nodes have been updated**

If there are rejected nodes, use the following mapping to determine the next step, based on the rejection reason

Rejection reason	Step name
[All reasons]	Index2
Bad quality scanned page	Scan
Wrong page sequence	Scan
Invalid OCRed value content	Index2

Note: The 'Step name' for specific reasons has higher priority over the 'Step name' of the [All reasons]

If there are no rejected nodes, the next step will be: Export

Figure 55. Review Functionality: branching Step

The checkbox will allow editing the table, the table contains two columns, the Rejection reason and the Step name, a step name can be defined for a rejection reason. If a user select any of the defined rejection reasons the Task will move to the defined step. The option for All reasons should always be defined, that is because it is possible to have a complex Workflow with multiple Rejection reasons that are defined to nodes in previous steps that are not known to the specific step. This option is available in order to handle such situations.

The steps below demonstrates some custom configurations and explains the available options for the Review availability,

Availability is set to: Enabled

The user is only able to review any rejected nodes and their rejected reasons without being able to change their status.

- For the Enabled *Review* configuration, see the sample configuration in the the image below:

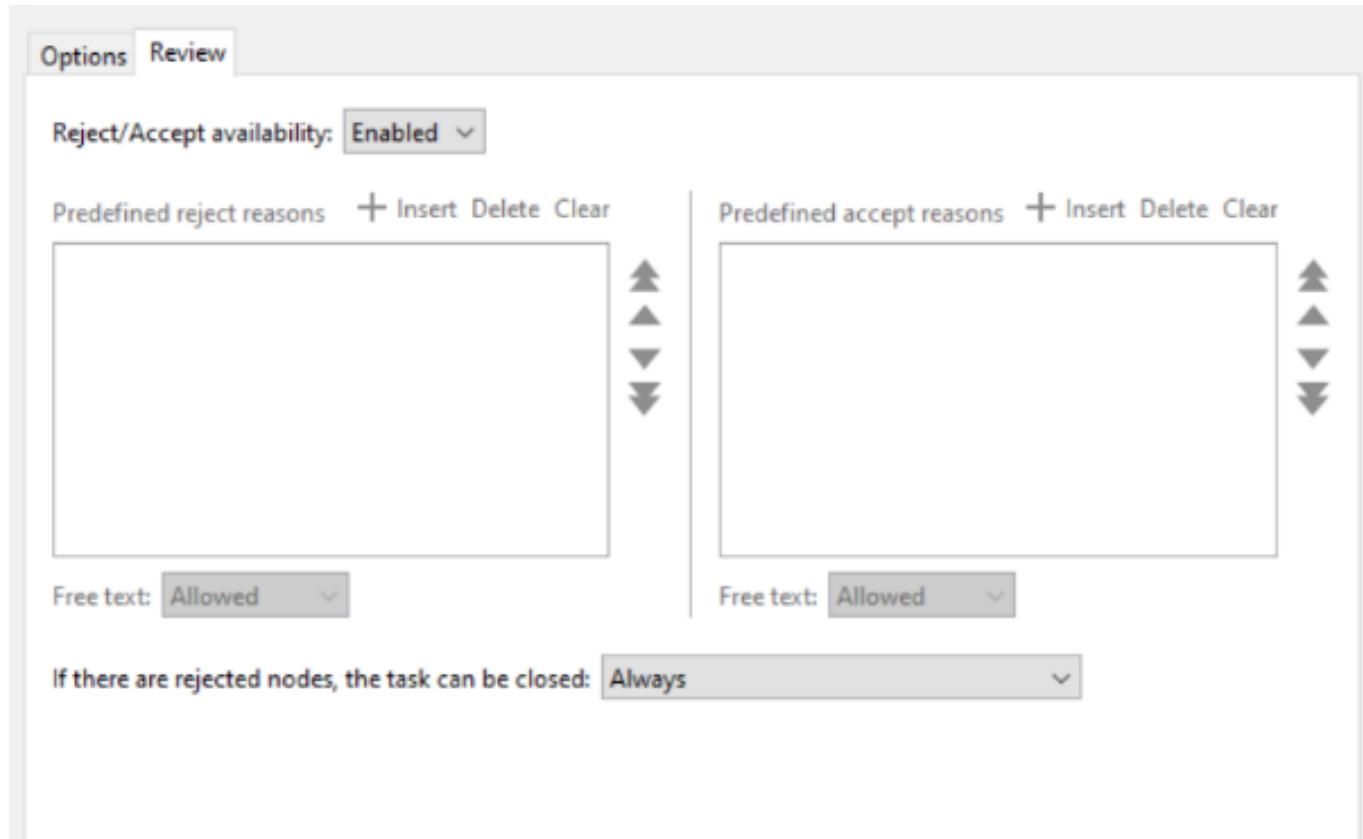


Figure 56. Review Functionality: Enabled Mode

- With the Enabled *Review* availability the tasks cannot turn to *Unrejected*. Therefore the option If there are rejected nodes, the task can be closed should be set to *Always* as selected above.
- The User can see rejected nodes at this step as shown below. On mouse over the rejected node, a tooltip with the rejected reason is showed, the right click context menu will also show the *Reject* /*Unrejected* history.

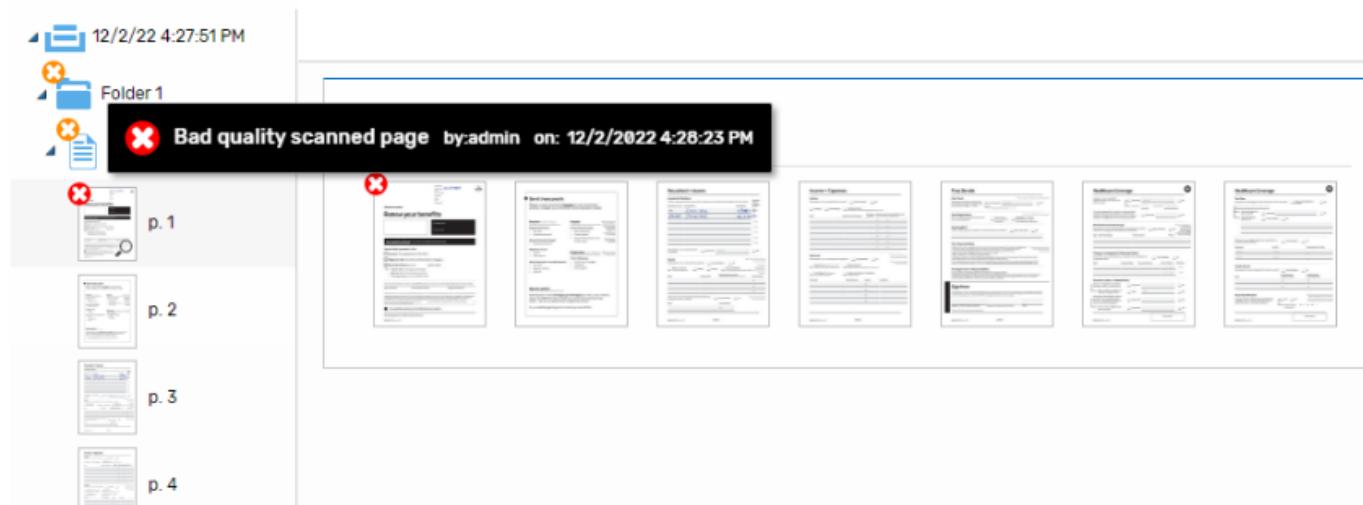


Figure 57. Review Functionality: Enabled Mode Client Side

Availability is set to: Editable

The User will be able to reject or unrejected nodes. If the *Review* step is a branching step, the Reject reasons can be mapped to a Workflow step that the task will follow.

For the Editable configuration see the sample configuration in the image below:

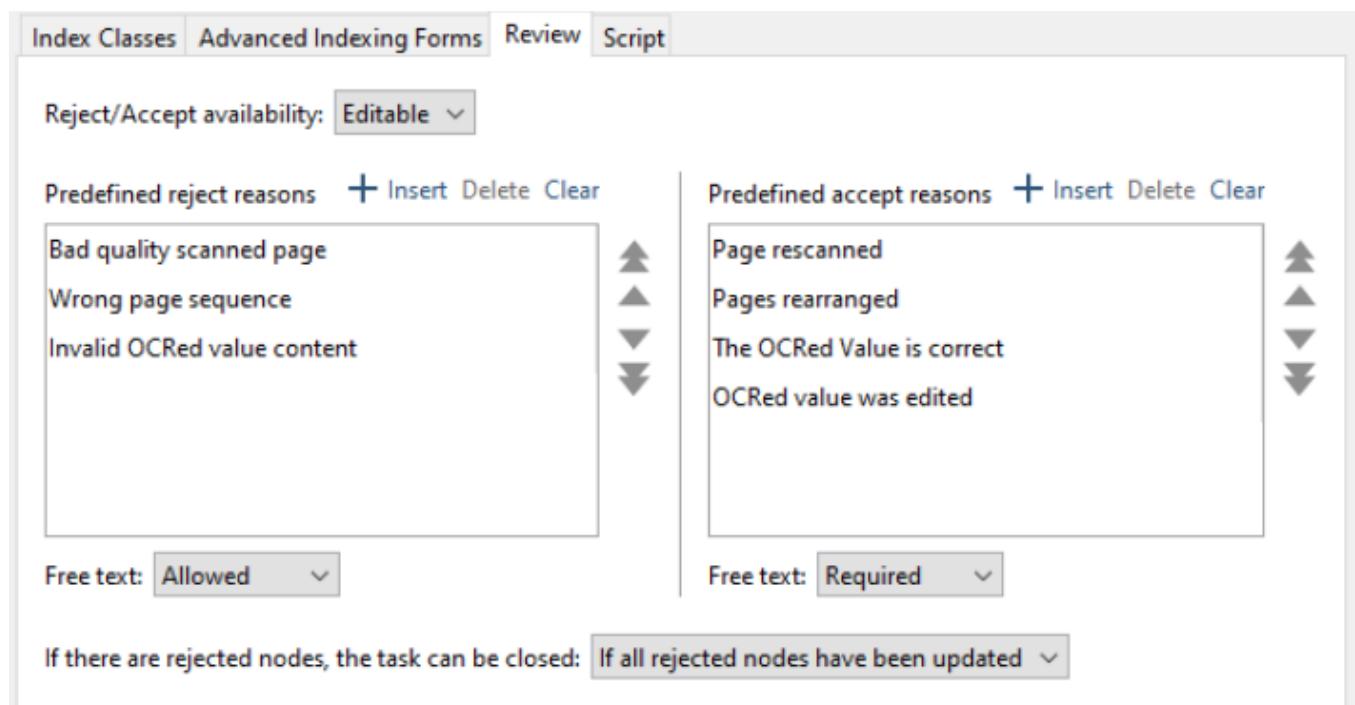


Figure 58. Review Functionality: Editable Mode

Here is the Workflow related to the above mapping:

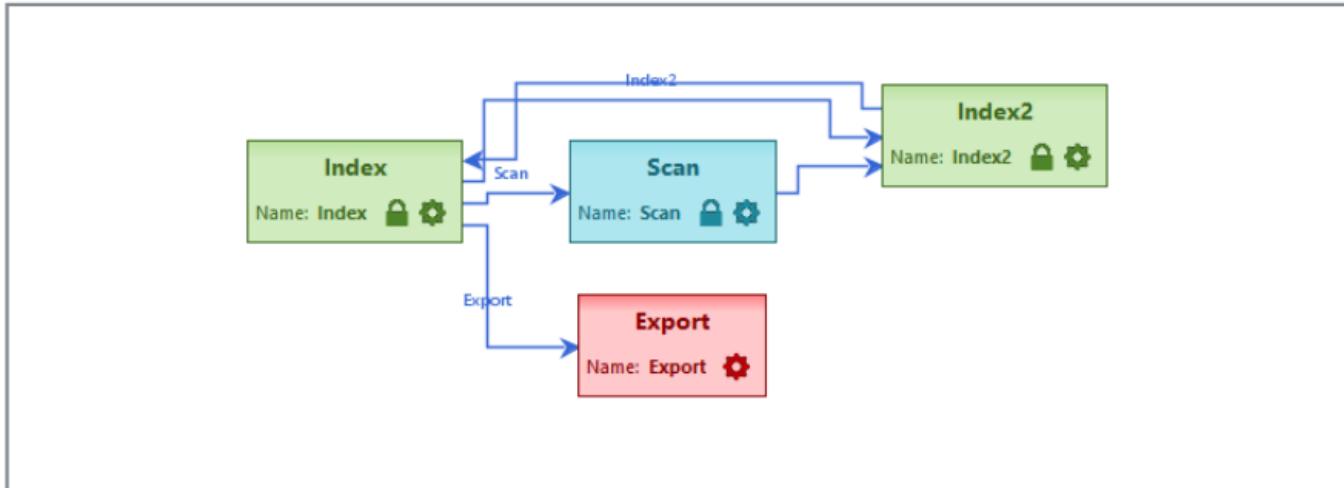


Figure 59. Review Functionality: Workflow

The Steps below demonstrates how to Reject a Document.

1. The User can reject nodes with right click then *Reject* page (or any node). The predefined reasons will show up, if any, by selecting a reason the selected node is rejected.

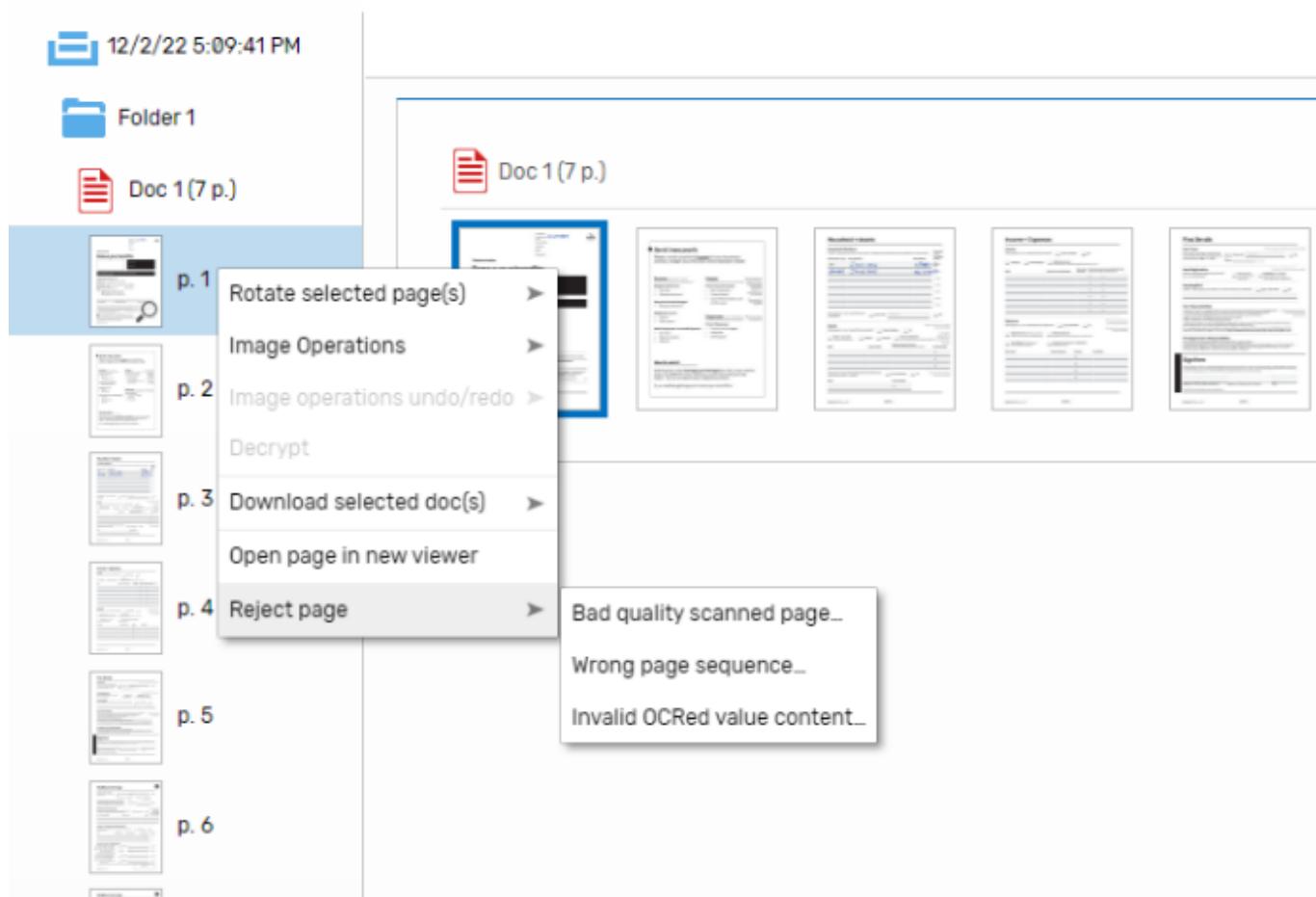


Figure 60. Review Functionality: Reject Step 1

1. After selecting a *Reject* reason it is possible to add a comment, Since the Free Text option was set to Allowed.

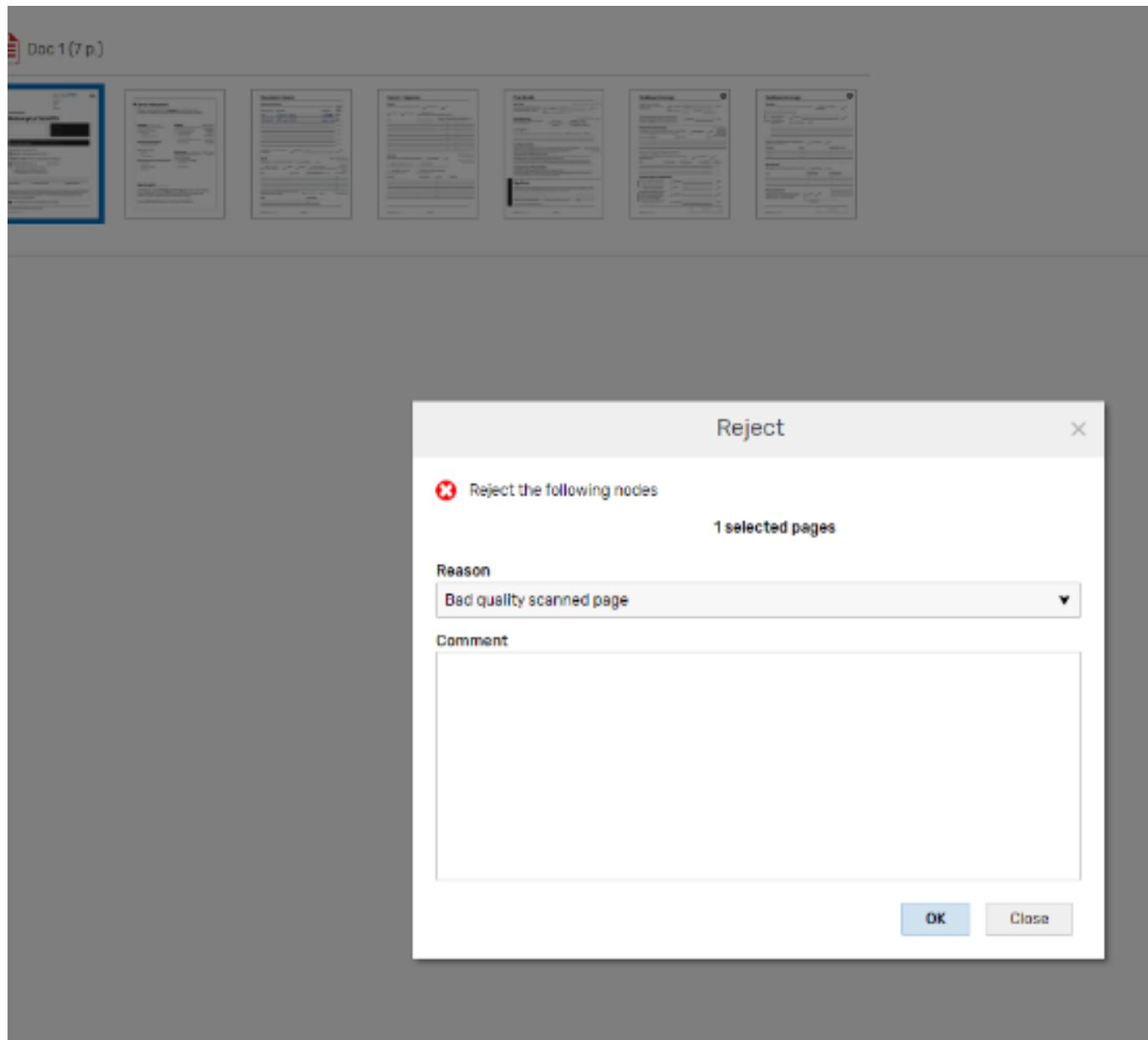


Figure 61. Review Functionality: Reject Step 2

The steps below demonstrates how the User can Unreject nodes:

1. The user can press right click then *Unreject* page (or any node), then select any of the *Unreject* reasons,

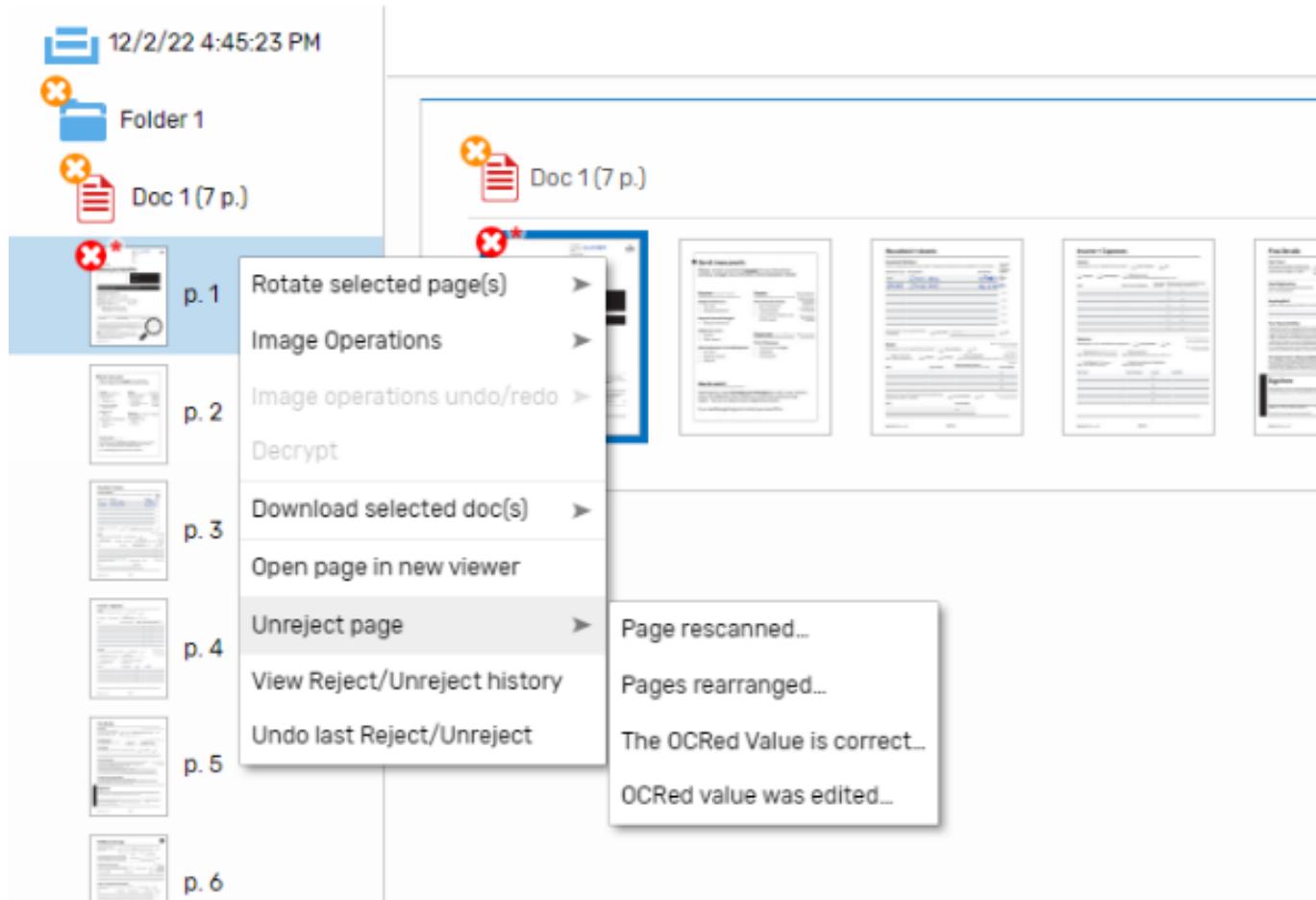


Figure 62. Review Functionality: Unreject Step 1

1. The node is unrejected once a reason is selected, in this step it is required to add a comment because the Free Text option was set to Required.

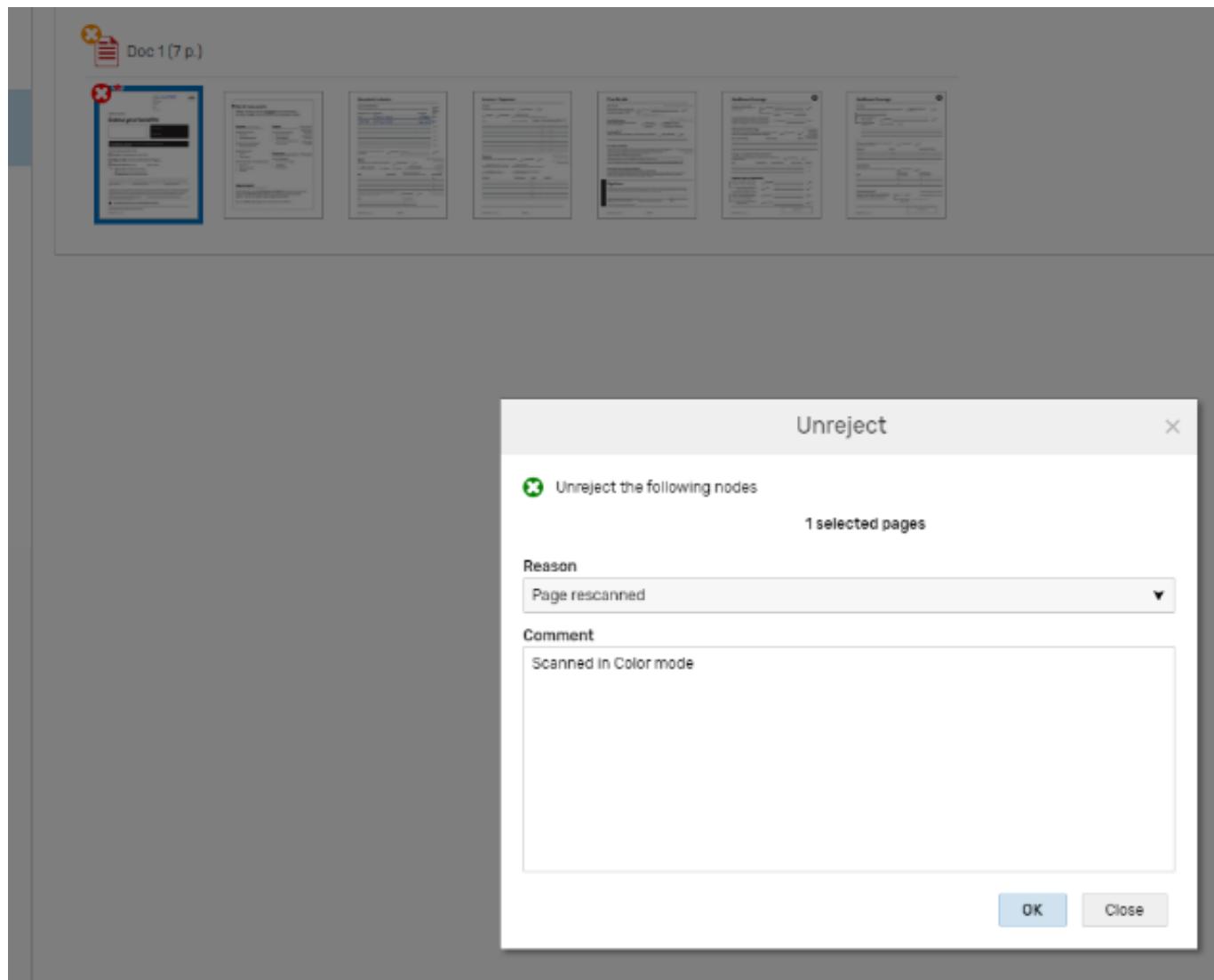


Figure 63. Review Functionality: Unreject Step 2

1. When the node is unrejected (node updated), it is marked as ticked and the task is able to be closed.

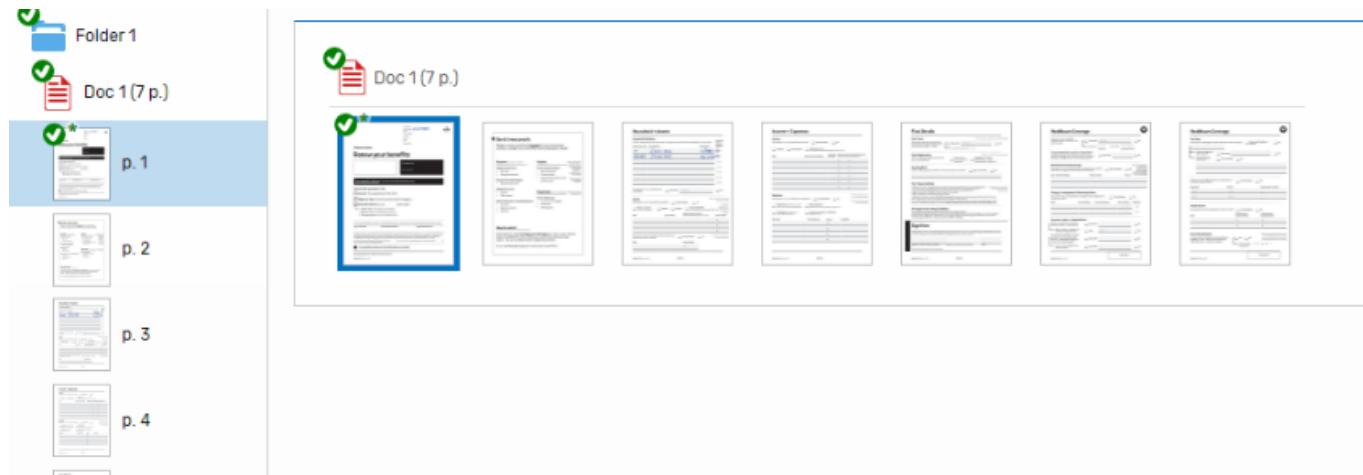


Figure 64. Review Functionality: Unreject Step 3

1. On mouse over the unrejected node, a tooltip with the *Unreject* reason and the related comment is showed.

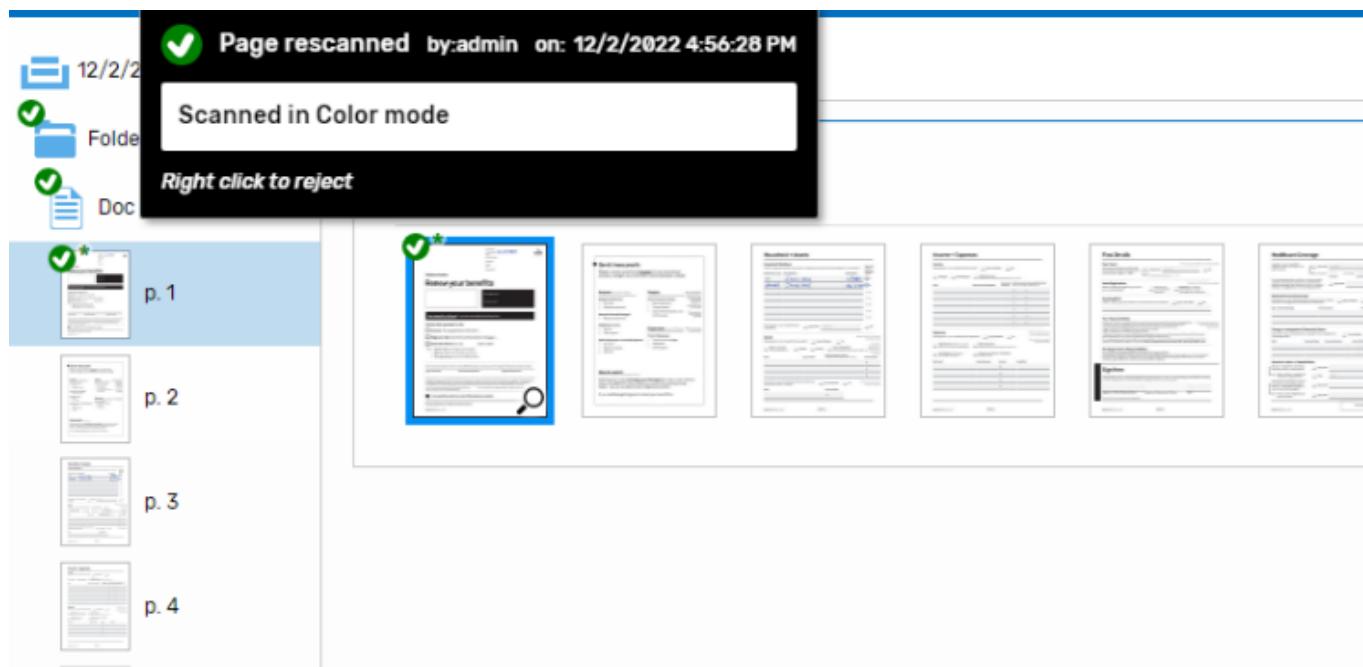


Figure 65. Review Functionality: Unreject Step 4

Availability is set to: Hidden

The will not be able to see any node rejections, or rejection history. For Hidden configuration see the sample configuration below,



Figure 66. Review Functionality: Hidden Mode

On *Hidden* the rejected nodes or the *Rejection/Unrejection* history will not be shown to the user.

3.1.11.11. Document Split

Document Split fundamentals

A server-side Document Split Job Workflow step is an automated document recognition module that leverages Machine Learning technology provided by third-party Cloud services to automatically identify specific document types and structures, such as invoices and standard forms and/or perform document separation. This step, in terms of functionality, can be compared to the Classification module that already existed in previous Info Input Solution versions. The difference lies on the fact that the Document Split module is fully automated and does not require any internal configuration or training.

Benefits from using a Job Workflow with a Document Split step

- Design your Workflow faster and with less effort.
- Quickly classify and/or separate documents.
- Ensure accurate and compliant results based on the biggest Cloud Service providers in the industry
- Automate and validate your data to make your workflows more efficient.
- Automate data capture at scale to reduce document processing costs.

- Make better and faster decisions using document data.
- Rely on the biggest Cloud Service providers' security models and world-scale infrastructure to keep your organization secure.

Setting up a project

Setting up an automatic Document Split project requires several configuration steps. A Document Split Workflow step can be combined with all different Workflow step types provided by the application, even with Intelligent OCR, Classification and Extraction steps.

The most common use case of Document Split is the following:

- Classify and/or separate documents, such as invoices and forms, which will be then proceed to an Intelligent OCR or Extraction step for OCR.

Step 1: Understand the content that will be handled

The first step should be to understand what content will be handled by the Job, and specifically from the Document Split Workflow Step, and to gather the requirements for the project. That includes: Getting a list of all types of documents that will be handled by the Job. The Cloud engines provide a list of pre-defined document types, selected as the most common use cases in the business.

For example, some of the document forms that the Google Document AI engine offers:

- 1003
- 1040
- 1065
- 1099
- 1120
- W2
- W9
- Account Statement Bank
- Payslip
- US Driver License
- US Passport
- Mortgage Statements

Step 2: Create the corresponding Document Class(es) or Form Type(s) in the Info Input Solution Job Setup.

The retrieved document types from a cloud engine must be assigned to a specific Document Class or Form type inside the Info Input Solution Job Setup. So, it is mandatory, before proceeding to the Mapping configuration, to have all these created. The creation of a Document Class is explained [here](#).

Step 3: Setup a Document Split Workflow Step

The Document Split Workflow step is available during the Job configuration.

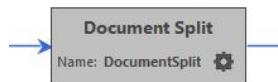


Figure 67. Document Split Workflow Step

Document Split Workflow Step

The fastest way to setup a Workflow containing a Document Split step is by selecting one of the present Workflows, by pressing the Change Workflow button. Of course, the Workflow can also be edited, in order to be combined with other steps.

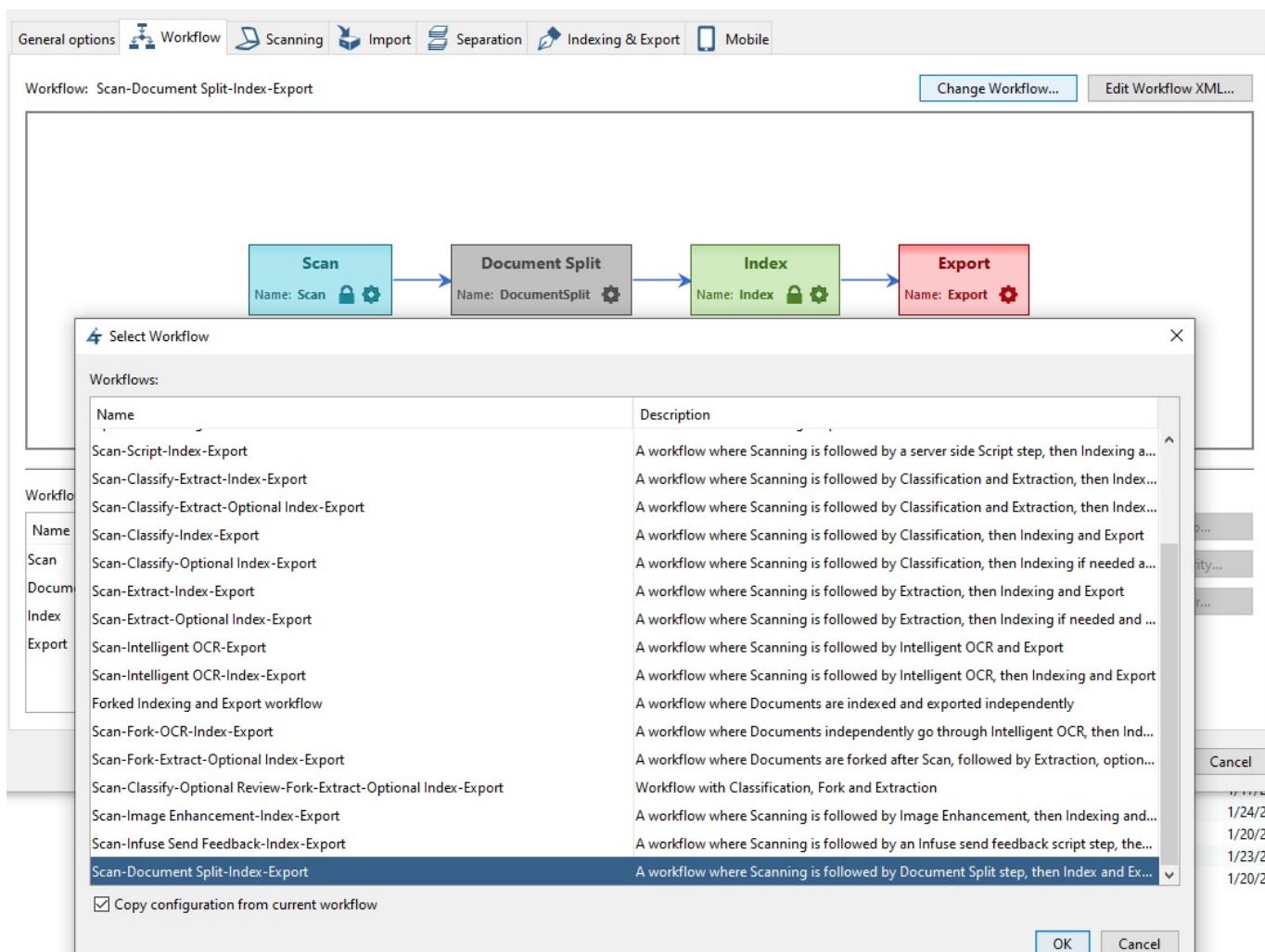


Figure 68. Job Workflow with a Document Split step

Creating a Document Split Project

General configuration

The configuration of the Document Split step is the most important part of a successful automatic document recognition workflow. A correct Document Split process means less manual Classification. To achieve this, you need to clarify the exact type of document that is going to be processed. According to this condition, a recognition profile that fits the business requirement needs to be created. To open and edit the Document Split configuration window, click on Edit Step button or the gear icon on the Workflow grid.

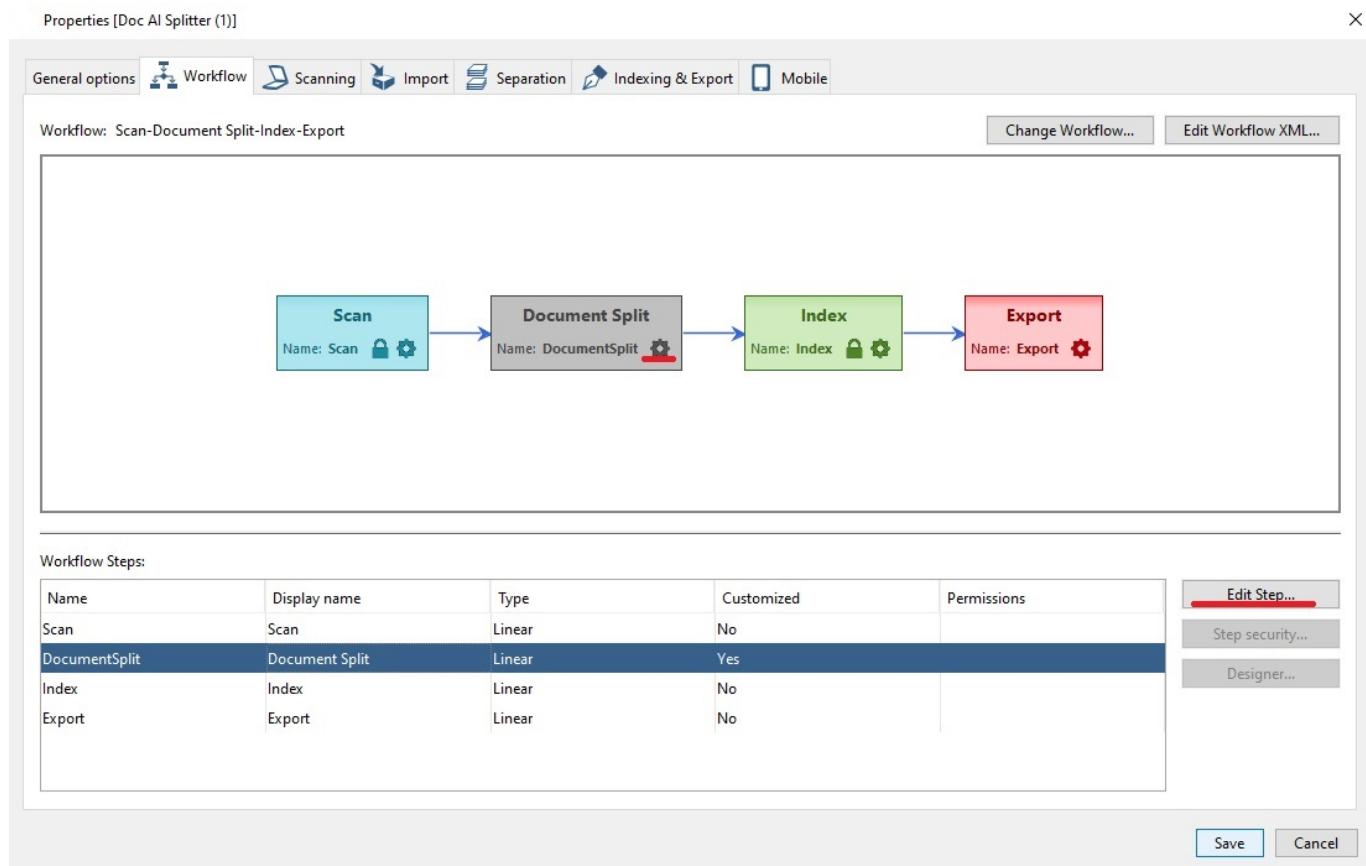


Figure 69. Edit the Document Split Workflow Step

Edit the Document Split Workflow Step

Options

The first tab of the opened configuration window is about selecting the appropriate Recognition profile according to the business requirement and configuring the basic options.

Workflow Step Configuration X

Workflow Step:

Type: Linear
Name: DocumentSplit
Display name: Document Split
Activity: com.imagetrust.docsplit

Options Error options Document type mapping Script

Extraction Profile: Document Split: Google DocumentAI Document Split Edit Extraction Profiles...

Process: All Documents Documents **without** Form Type

Only process the first Pages of each Document

Page image selection:

Default image Image with alias: Var ▼

If image with alias does not exist: Use default image Stop processing (error)

Perform the following actions:

Split Document
 Classify Document
 If the Document has been processed before, classify again
When confidence is at least %

Execute custom post-processing script function

OK Cancel

Figure 70. The Document Split Options tab

The Document Split Options tab

1. Extraction Profile: Select the recognition profile that will be used to connect to the Cloud engine and Classify documents.
2. Process: Select whether the Document Split module will run for all the documents in a batch or only for those that do not currently have a Form Type assigned. Moreover, it can be run only for a specific number of pages in each document.
3. Page image Selection: This option applies only in case an Image Enhancement Workflow step exists prior to the Document Split step. The Default image option means that the originally imported image will be used by the Document Split step. The Image with alias option means that the Document Split step will use the image that was produced by the Image Enhancement step.
4. Perform the following actions: Split Document: On successful identification, a new document will be created, starting from this page. Classify Document: On successful identification, classify and assign

a Form Type to the identified document. If the Document has been processed before, classify again: If there are already data from a previous Document Split step, decide whether the image will be processed again. When confidence is at least: Decide what the minimum accepted confidence from the Cloud engines will be. Execute custom post-processing script function: The **postProcess** function is placed inside the Script tab where custom code can be applied.

Error Options

In the second tab, an Administrator can decide what happens in case an error occurs during the process.

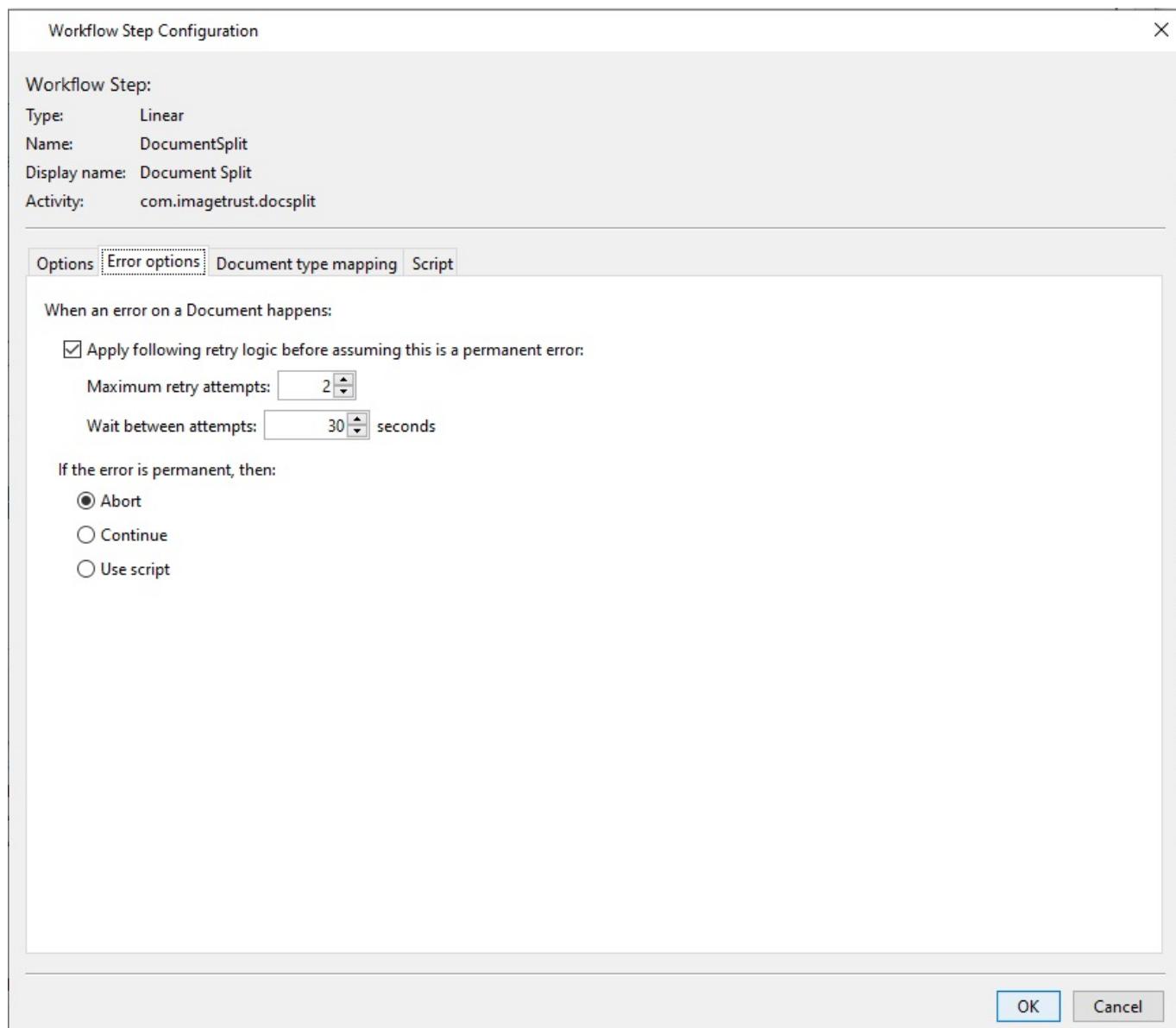


Figure 71. Document Split Error Options

Document Split Error Options

The error policy is applied in this step. The configurable settings are the number of the retry attempts and the time threshold between them. If the error is permanent, it is decided whether the process will be aborted (and the task will be moved to Error state), or the process will be continued without assigning a Form Type. The last checkbox will use the scripting logic that is applied in the last tab, when a permanent error occurs.

Document type mapping

In the third, and most critical tab, the mapping between the results from the Cloud engines and the Info Input Solution Document Classes or Form Types will be configured.

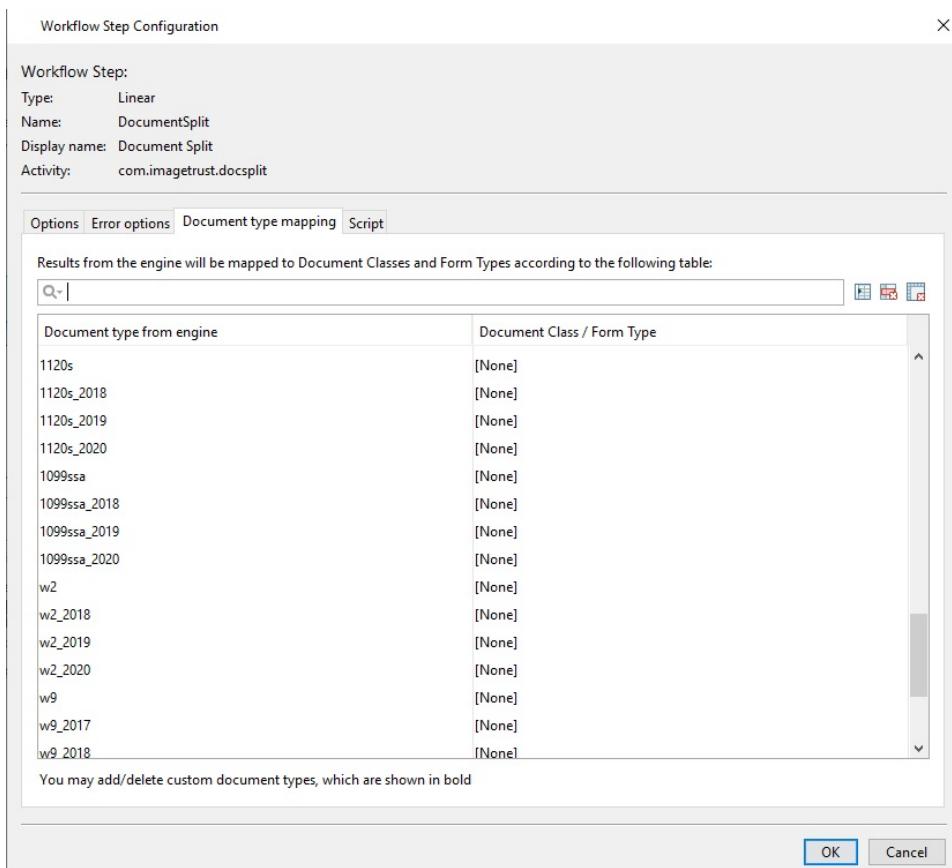


Figure 72. Document Split Mappings tab

Document Split Mappings tab

The left column of this table (Document type from engine) will be filled with pre-defined values once a valid recognition profile is selected in the Options tab. This is the list of values offered by the specific Cloud engine. In the screenshot above, the Google Document AI Doc Splitter is used. For document types that are not in the pre-defined list, the Add row button from the top right corner of the table can

be used, to add custom document types. Of course, the custom types must be offered by the engine.

When using the Microsoft Azure Recognizer Custom Classifier for Document-Splitting, the Document type must be added manually by using the Add row button.

The right column of this table corresponds to the Document Classes and Form Types that are associated with this specific Job Setup. Multiple document types from the left column can be mapped to a Document Class or Form Type from the right column.

Script

In this tab, the `onError` and the `postProcess` functions can be found. Both of them have a detailed description and comments on when they are triggered and how they can be used during the process.

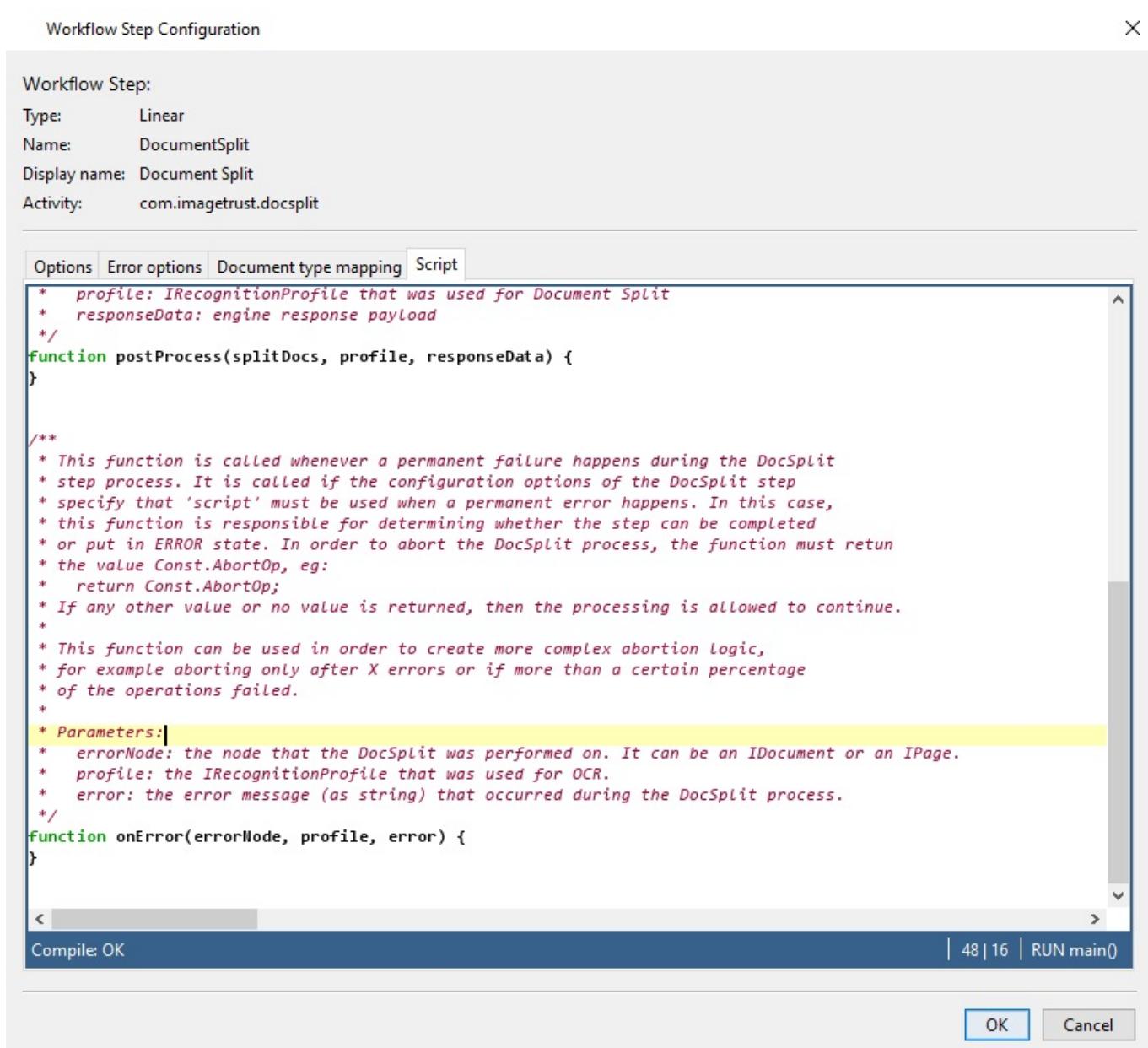


Figure 73. Document Split Script tab

3.1.12. Indexing

Info Input Solution is a *Batch* processing system, yet the central element of processing is the *Document*. A *Document* in *Info Input Solution* is the representation of a physical paper document (like an invoice or a contract) and consists of one or more pages. Indexing is the process of classifying *Documents* and assigning index data to them. Each piece of information is saved in an *Index Field*. For example, an *Index Field* can be the first name on a contract, the last name, or the full name. Usually, the definitions of *Index Fields* are such so that when the *Documents* are exported along with their metadata to a permanent storage (e.g. a Document Management system). Field definitions should match the fields that are

defined in the permanent storage.

Each *Index Field* has a type, called *Field Type*, that defines the allowed values. *Field Types* in *Info Input Solution* match common database field types, like CHAR, DATE, INTEGER, DOUBLE etc.

A *Document Class* is the definition of a specific type of document, and includes the specification of:

- a set of *Index Fields*, along with their types and specific properties
- one or more *Database Actions*, that are executed when the values of an *Index Field* change
- a JavaScript script where you can fine-tune the run-time behavior of the *Index Field* values
- a set of *Export* configurations for this type of *Document* when processing ends

Document Classes are *Shared* objects in *Info Input Solution*, and may be linked to one or more *Jobs*. You can create any number of *Document Classes* and then link one or more of them with one or more existing *Jobs*.

Documents may be grouped in *Folders* for organizational purposes. *Folders* themselves can also be associated with a set of *Index Fields* and be exported along with *Documents* and pages. A *Folder Class* is thus similar to a *Document Class*, since it includes the same set of data (*Index Fields*, *Database Actions*, JavaScript script, Export configurations), but applies to the folder level. *Folder Classes* are also *Shared* and linked to *Jobs*.

In *Info Input Solution*, you can also define the same set of data (*Index Fields*, *Database Actions*, JavaScript script, Export configurations) to the *Batch* itself, but there are two differences with *Folder* and *Document Classes*:

- There is no reusable "Batch class" - these 4 sets of data are directly defined on the Job
- Since there is no "Batch class", these data are not linked to the Job but are rather a part of the Job definition: this means that they are not reusable for other Jobs

3.1.12.1. Index Fields

An *Index Field* represents a piece of information (like a name or an address or a date) that comes from a document, or represents a property of the system (like the scan date), and can be exported at the end to a permanent storage.

Index Fields are used to store information during the life-cycle of a batch, as it moves through *Info Input Solution*. You can think of *Index Fields* as properties associated with the different entities of a batch. *Index Fields* can be associated with the *Batch* itself, with *Folders* and with *Documents* (not with pages).

You can define the set of *Index Fields* that you want to associate with each level of a *Job* (*Batch*, *Folder*, *Document*) via *Batch level indexing*, *Folder Classes* and *Document Classes*.

During indexing, you select the class of each *Folder* and *Document*, and following this selection, each one *Folder* and *Document* inherits the *Index Fields* that have been associated with the corresponding *Folder* and *Document Class*.

You can create/edit *Index Fields* from the *Index Fields dialog* which is accessible from several paths:

- Batch level Index Fields: *Setup data dialog* → *Jobs tab* → *Job Properties dialog* → *Indexing & Export Tab* → *Define index fields...*
- Document Class Index Fields: *Setup data dialog* → *Document Classes tab* → *Index Fields...*
- Folder Class Index Fields: *Setup data dialog* → *Folder Classes tab* → *Index Fields...*

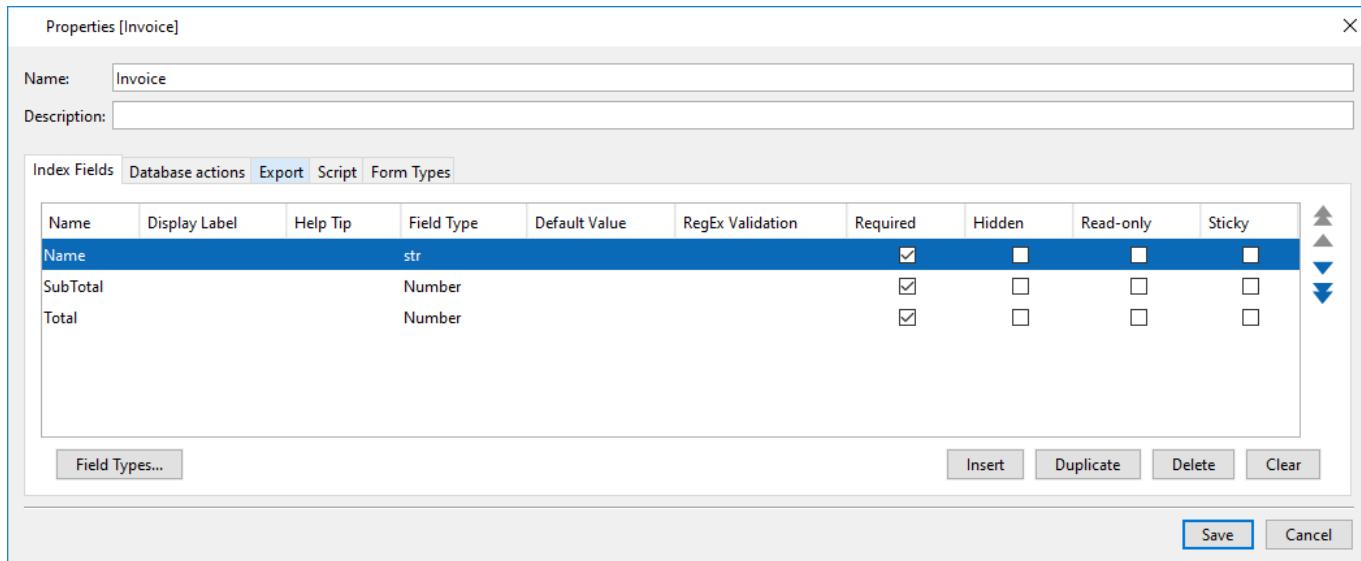


Figure 74. *Index Fields dialog*

Creating new Index Fields

- Click the *Insert* button to add a new *Index Field*.
- Double click on a *cell* in the table to *edit* the value of each property you want to set:
 - *Name*: this is the name of the *##field*: it should not contain spaces. You should only use **alphanumeric characters** and the underscore char (_). *Index Fields* are referenced in scripts by their *names* so it is suggested to use short and descriptive names.
 - *Display Label*: this is the label that will appear at the *Indexing Panel* to the user. If you leave it empty, the *Name* will be used
 - *Help Tip*: this is a multi-line tip that will appear at the *Legend area* of the *Indexing Panel* (and as a *tooltip*) during indexing: you can write something here to help the index operators understand what they should enter.
 - *Field Type*: this cell contains details about the data type of the field. You can click on the *Field types...* button to define new field types if required. To change the *Field Type*, click on the '...' button inside the *Field Type* cell. This will bring up the *##Define field type details* dialog:

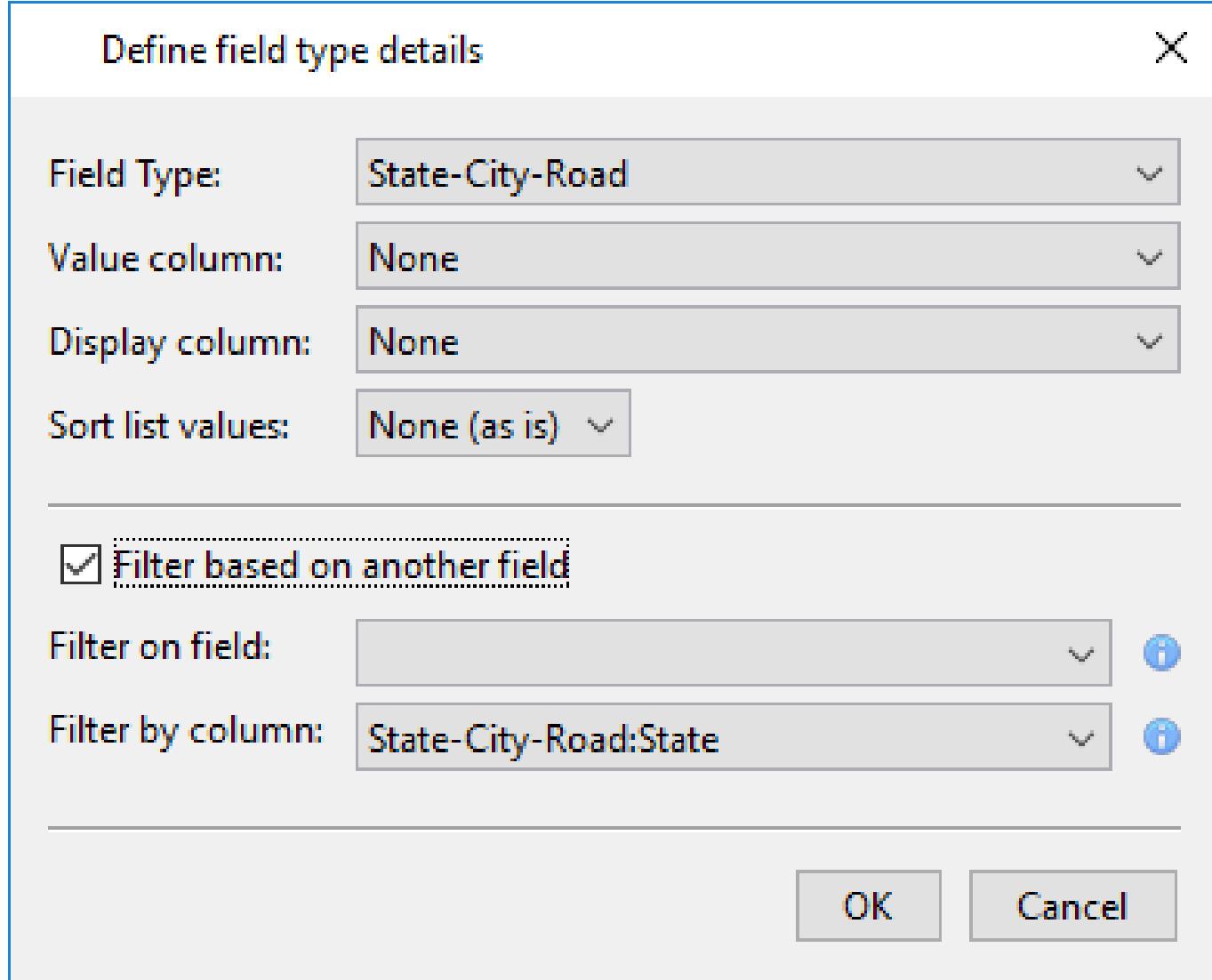


Figure 75. Define field type details dialog

In this dialog you may define the following

- **Field Type:** Choose the Field Type that this Index Field will use. This will determine whether the Index Field is of type string, number, date etc.
- **Value/Display column:** If the selected Field Type contains a list of possible values, then Value and Display column properties will be enabled and you will be required to fill in at least one of them. These determine which column of the Field Type table will be used to give to the Index Field its commit value and which column will be used to display the available choices.
- **Sort list values:** If the selected Field Type contains a list of possible values, then you may choose if these values will appear in ascending, descending order or in the order they are listed in the Field Type table.
- **Filter based on another field.** Check this box if you want this Index Field to modify its drop-down list of possible values, based on the values of some other Index Field.

- *Filter on field*: If the '##Filter based on another field' box is checked, then this property will be enabled and you will be required to choose the Index Field whose value will be used in order to filter the possible choices for this Index Field.
- *Filter by column*: If the 'Filter based on another field' box is checked, then this property will be enabled and it will be filled with the column names of the Field Type that is used by this Index Field. The filtering of the possible values happens by taking the display value of the Index Field selected in the *Filter on field* box and looking it up on the Field Type's table within the column selected in *Filter by column*.
- *Default Value*: this is a list of values with which the field will be initialized the first time it is displayed to the index user. See the [Variables](#) topic for possible values.
- *RegEx Validation*: you can insert here a *regular expression* that will be used to validate the user's input. A list of some predefined regular expressions is provided, mostly as a guidance to help you write your own.
- *Required*: if a field is required, the user will need to provide a valid value before indexing is considered complete.
- *Hidden*: if a field is hidden, the index user will not see it. Hidden fields are very useful to carry information that are invisible to the index user (for example, you can create a hidden field and set its default value to the `${OSUserName}` so you capture the Operating System logged in user). Hidden fields are accessible through scripting, without being visible to the user, thus making them very useful as pass-through variables attached to a batch/folder/document.##
- *Read-only*: if a field is read-only, the user will not be able to change its value (for example you can use a read-only field to give the index operator some system value to compare or check).
- *Sticky*: the value of a sticky field is copied over to the next node (folder, document) as indexing proceeds, so the user does not need to enter its value again, unless it changes
- Click outside the table to save your changes.
- You can use the arrow buttons on the right to re-arrange fields: the order you set here is the order that they will appear to the index operator.

3.1.12.2. Tables

A table is a group of fields for values that logically occur multiple times in the electronic form, for example, the Quantity column in a purchase order. Essentially the Table fields are containers of Index Fields, the Index fields are the columns of a table, when an Index Field is inside a Table then it can hold more than one value. The Tables are available in read write mode only in the *HTML Client*, in the *Thick Client* the Table fields are shown in a read only mode.

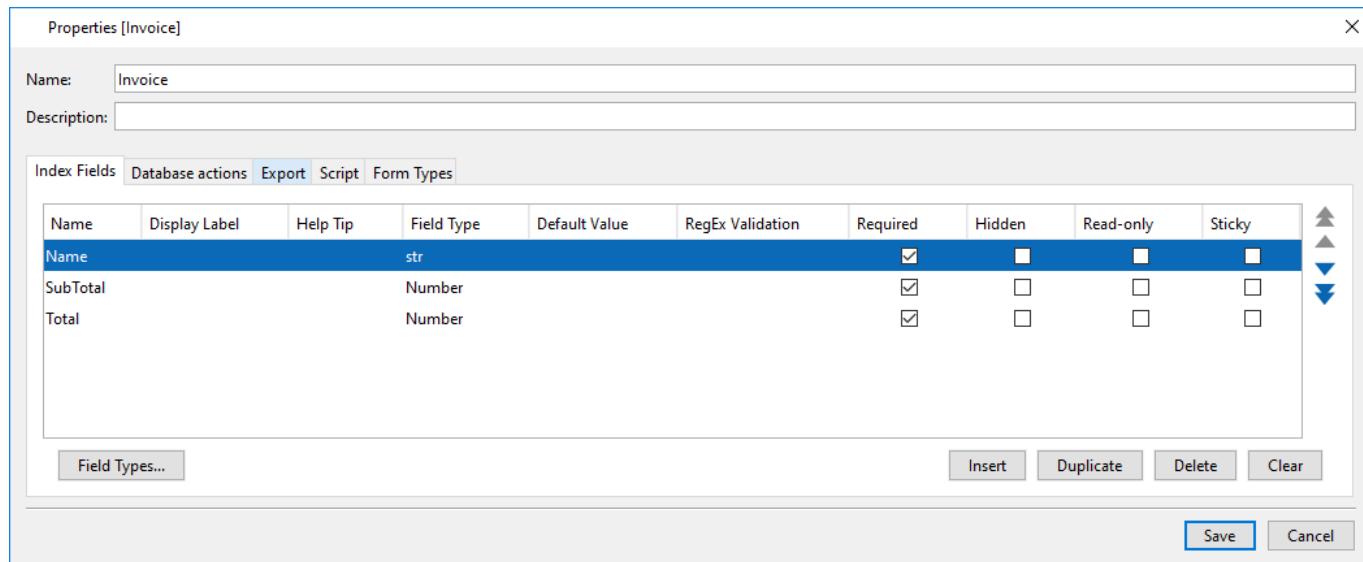


Figure 76. Index Fields dialog

To add a table select the Insert Table... button from the Fields configuration of the Index Class as shown above, then the following dialog will show up,

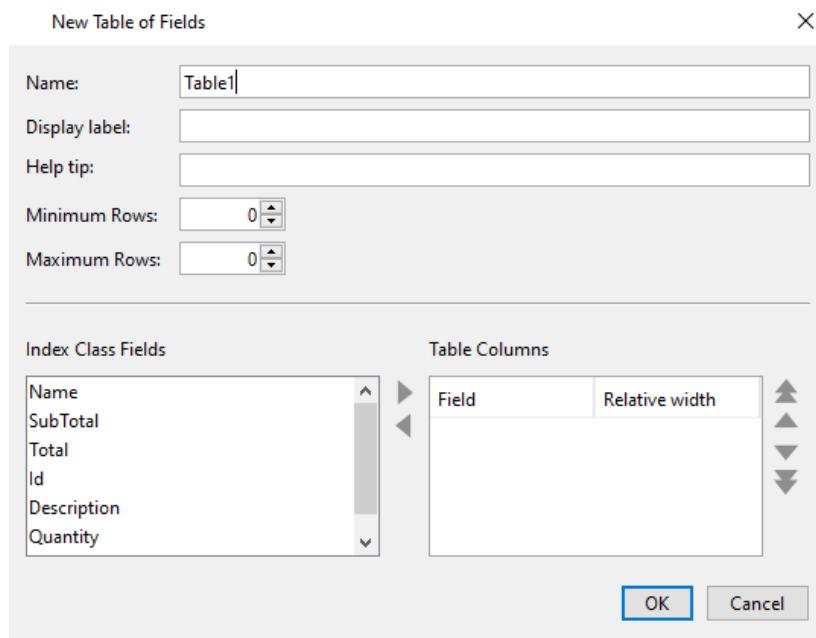


Figure 77. Table Field Creation

The available controls are the following,

- Name: this is the name of the table, this will show up in the Index configuration and in the index panel at the client GUI, in case the Display field is empty. Any scripting reference to this table will be done with this name.

- Display label: The display label holds the name that will be shown for the table, this will only show in the GUI and it does not have any internal reference to the scripting.
- Help tip: The help tip can be used to hold some business specific information about the table field.
- Index Class fields: This table holds all the Index Fields that are available in each Index Class and can be used inside the table. The Index Fields that are inside this table are not selected for this Table.
- Table Columns: This table Holds the Index Fields that are selected for the specific table.
- Relative width: This property will set a predefined width for the columns of the table.

Select OK to add the Table field in the Index Class

3.1.12.3. Form Types

A *Form Type* defines a unique form of a type of document; for example, a *Document Class* representing an invoice may define a set of fields pertaining to the invoice (e.g. *name*, *address*, *total*), and contain two different *Form Types* that represent the invoice from two different vendors. Although both invoices have the same fields, the layouts of the invoices may be different, so each *Form Type* is used to represent the invoice from each vendor. Each form type can have associated *sample pages* of the actual paper document it represents; moreover it supports the definition of *Index zones* where the index fields are physically located on the physical paper. It is thus possible to customize the processing of each different invoice, although the actual type of the document (along with the meta data) is the same.

Form Types always exist in the context of a Document class. You create *Form Types* from the *Form Types* dialog, which you can access from the *Setup data dialog* → *Document Classes tab* → *Form Types...*

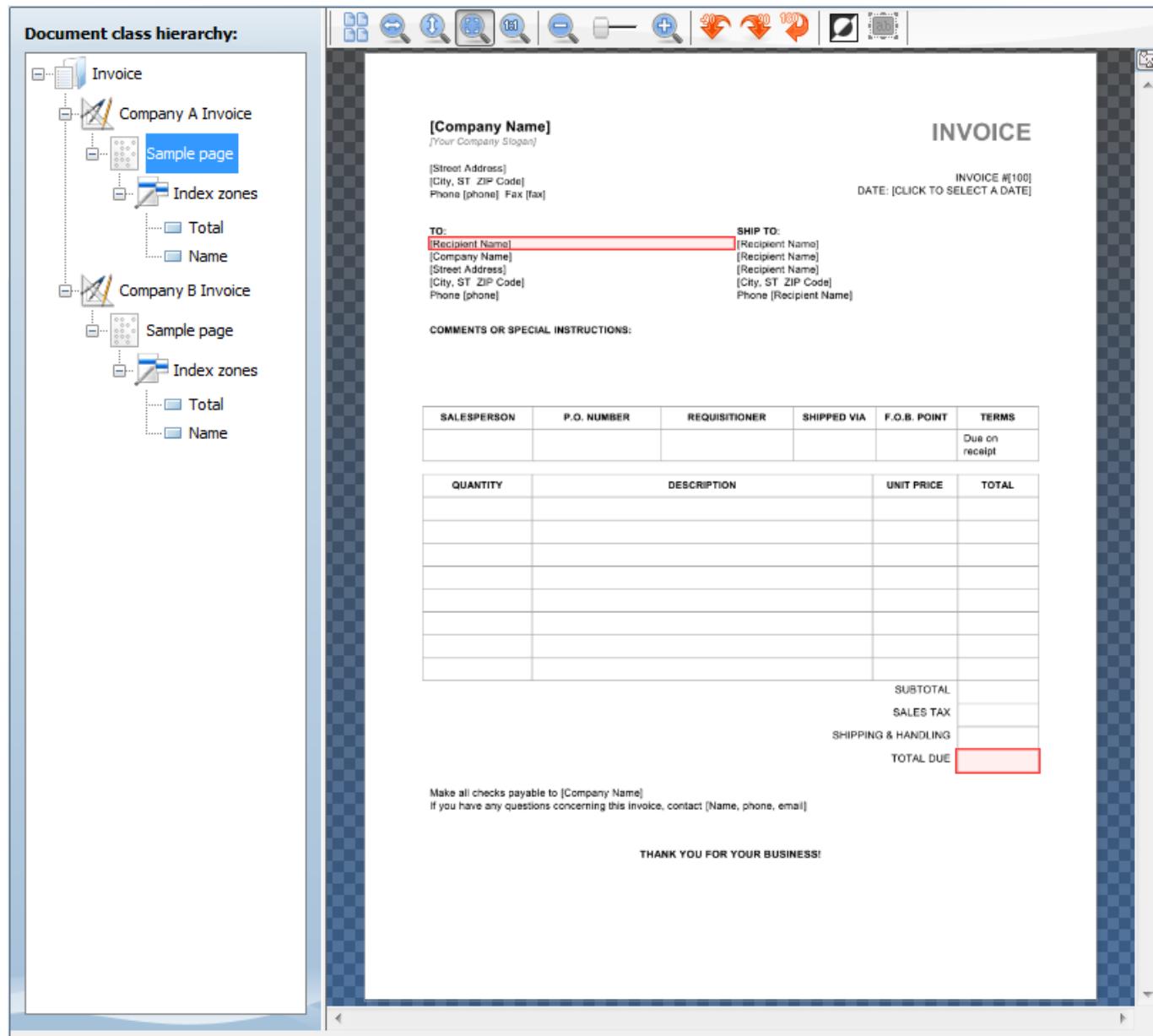


Figure 78. Form Types dialog, displaying form type Company A Invoice with an invoice sample

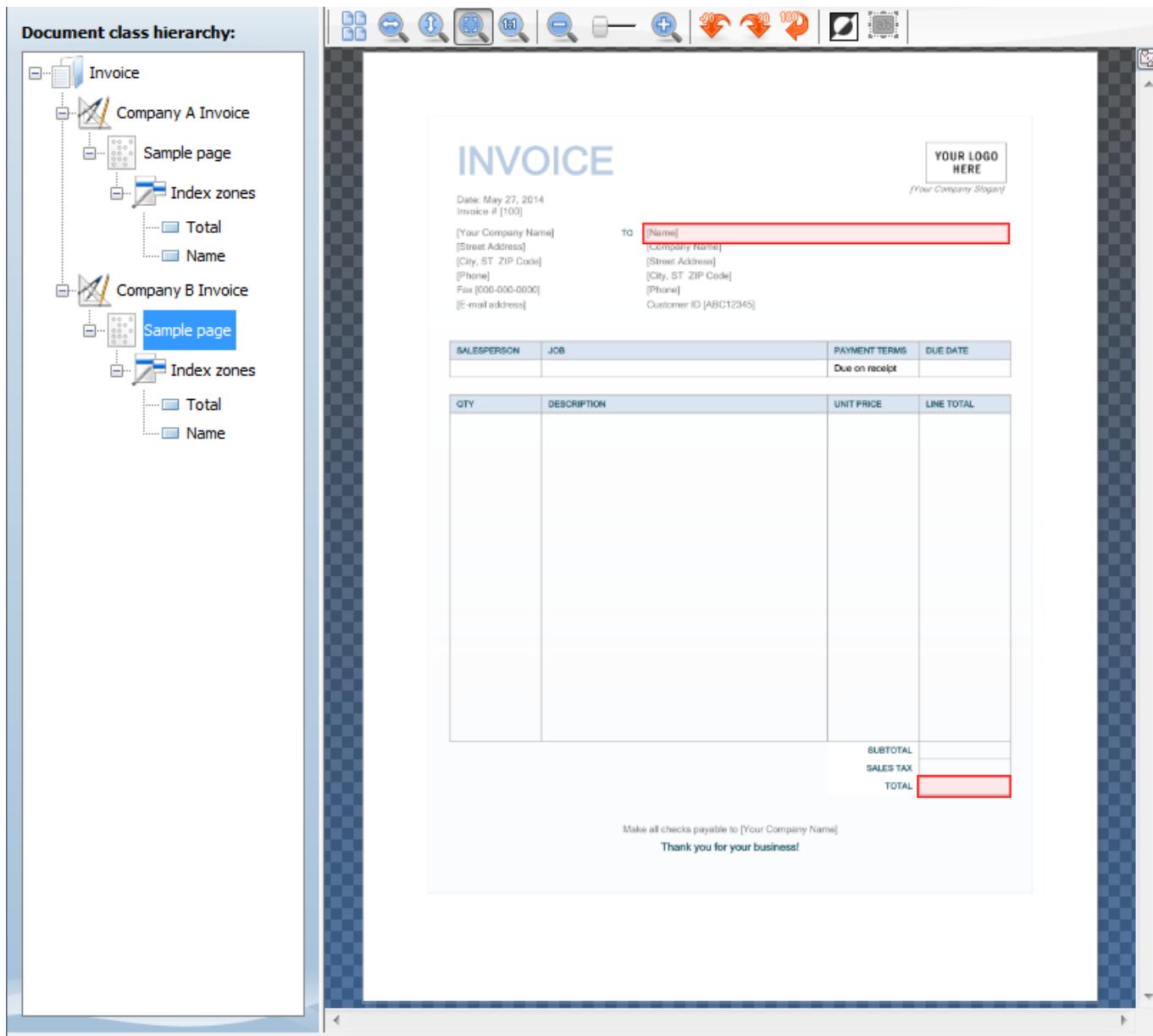


Figure 79. Form Types dialog, displaying form type Company B Invoice with an invoice sample

In the two images above you can see the *Form Types* dialog for a *Document class* named *Invoice* with two fields defined, *Name* and *Total*. Two *Form Types* have been defined that represent invoices from two different vendors: the *red rectangles* on the forms are *Index zones* that correspond to the two fields: notice that for each form type, the zones are in a different area.

To create a new *Form Type* right-click on the first-node on the tree and select *New Form type...*

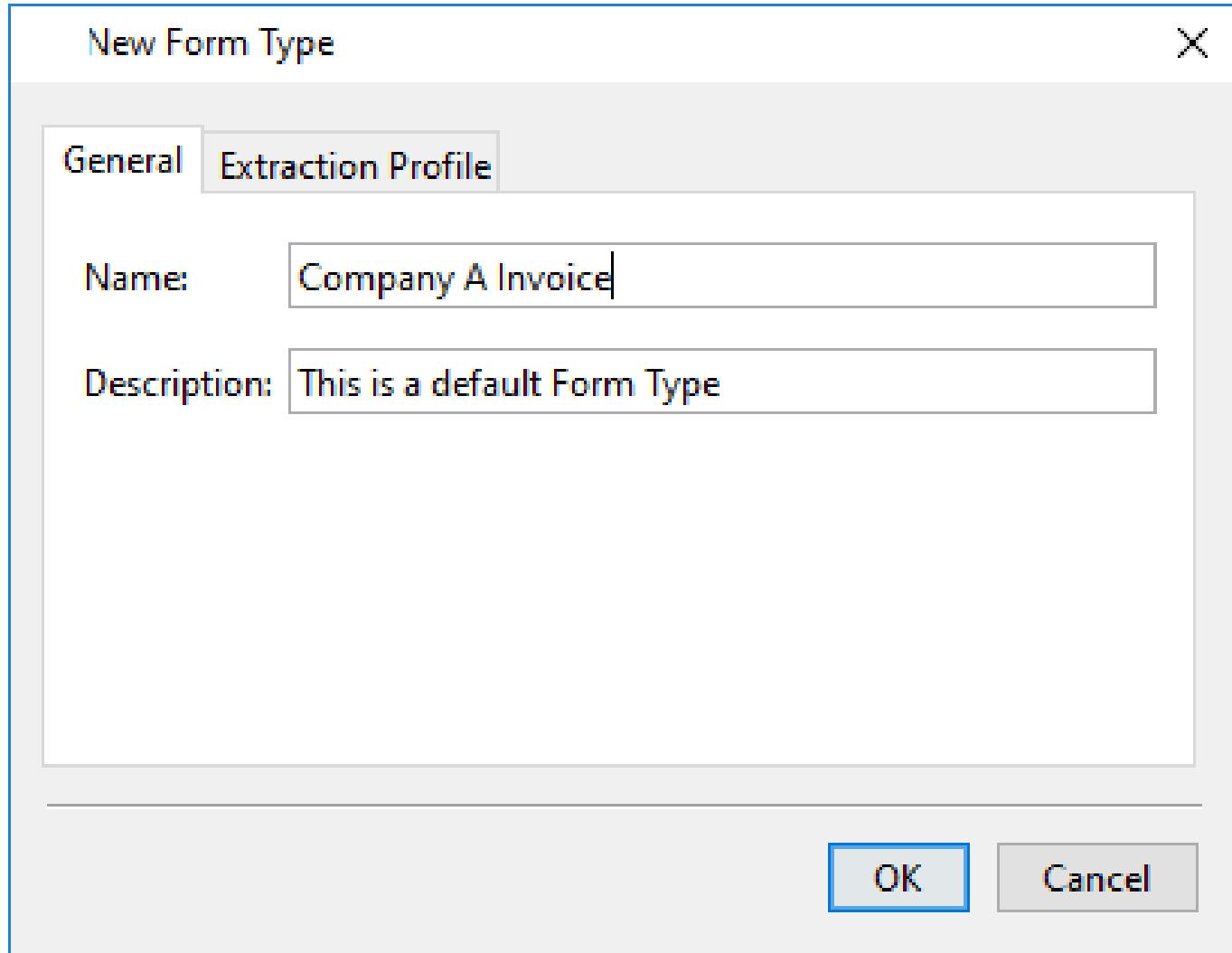


Figure 80. Form Type properties dialog

In the *New Form type* dialog, fill in the *Name* and *Description*. You may select an *Extraction Profile* for the *Form Type* if you need to extract some page-level barcode or patch code from the specific form.

Sample Pages

Info Input Solution allows you to add a sample of the actual document that users will be scanning: these are called *Sample Pages*.

To add a *Sample page* to a *Form Type*, right click on the *Form Type node* on the tree and select *Add sample page*. Each Form Type can have multiple *Sample pages*, since form types represent actual documents which may have multiple pages.

Thumbnails of the *Sample Pages* are presented to the operator during indexing to help the user select the correct *Form Type*. If a *Form Type* has more than one Sample pages, the first page is used as a thumbnail during indexing.

Index Zones

To add an *Index zone* on a *Sample page*, right click on the *Sample page node* on the tree and select *Add new index zone*. When you do so, the mouse cursor will change to a cross: you can then *drag your mouse* over the sample page on the right-area while holding down the left-mouse button to create the *index zone*. Once you export your mouse button, the *Index zone properties dialog* will appear:

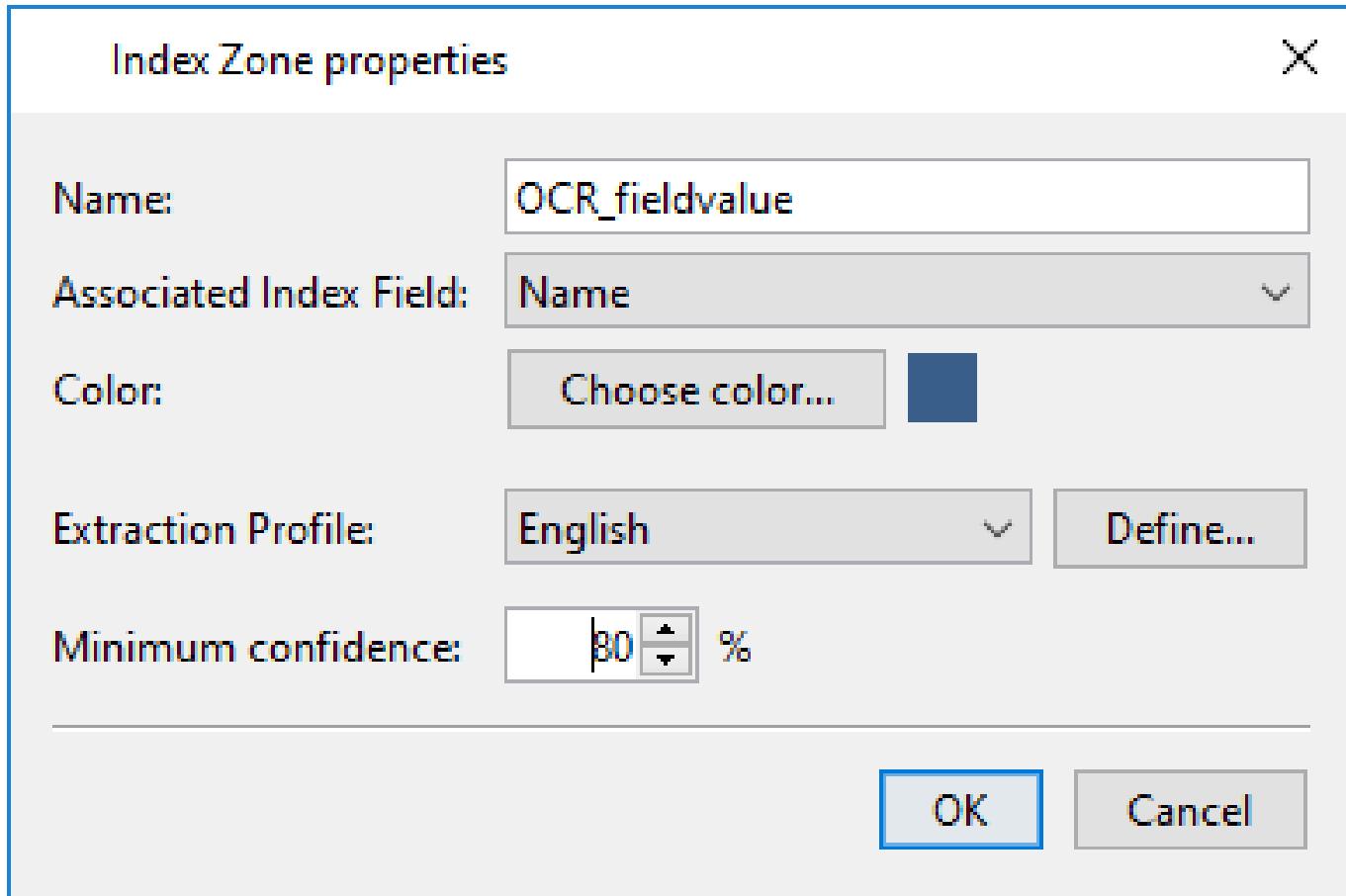


Figure 81. Index zone properties dialog

From this dialog you need to select the *Associated index field* that this zone represents, and the *Color* of the rectangle on the page that you want (this color will also be used during Indexing). Then, you can define the *OCR Extraction Profile* that will be used during Indexing. The *minimum confidence* is a threshold value that will be compared with the confidence metric, provided by the *OCR engine component*. When the *OCR output* has a lower value than this threshold, the *index field value* will be marked as *Invalid*. In this case, it will be required for the operator to review this *index field*, before closing the batch.

If the *Index fields* you have defined reside on the second, third, etc page of the form type (provided you have added 2, 3, etc sample pages), then when the *Index field* is focused during Indexing, the *Image viewer* will display the corresponding second, third, etc page of the actual scanned document.

During indexing, *Info Input Solution* displays in a separate window, called *Zone Zoom pane*, an enlarged view of the area that you define in the index zone during this step.

Editing Index zones

You can edit existing *index zones* either by:

- right-clicking on the index zone in the viewer.
- right-clicking on the node on the tree on the left that represents the index zone.

In both cases, you get a menu with *Properties*, *Duplicate* and *Delete* functions.

If you single left-click on an *index zone* in the viewer, four *handles* will appear on the corners, which you can use to resize the zone; you can move the zone around by dragging it with the mouse.

3.1.12.4. Database Actions

Database Actions (abbreviated *DB Actions*) are generic database operations that can be performed while indexing in order to:

- validate the values of *Index Fields*
- auto-fill the values of *Index Fields* based on the value of another *Index Field*
- perform some other task in a database by executing an arbitrary database command

Using a *Database Action* increases the speed at which users can index and eliminates typing errors. For example, a user might enter the first few letters of a customer's company name and perform a database lookup. The system would search a database for records with company names starting with those letters and display a hit-list. After the user chooses the appropriate record from the hit-list, the system will populate the appropriate *Index Field(s)* from the database record.

All database queries are executed from the Core Service, not the Client. This means that connectivity to the database must be ensured from the server node that hosts *Info Input Solution*.

Database Actions are associated with *Index Fields* and are defined for [Batch level indexing](#), [Document Classes](#) and [Folder Classes](#) by clicking the *Database Actions...* button or popup menu item:

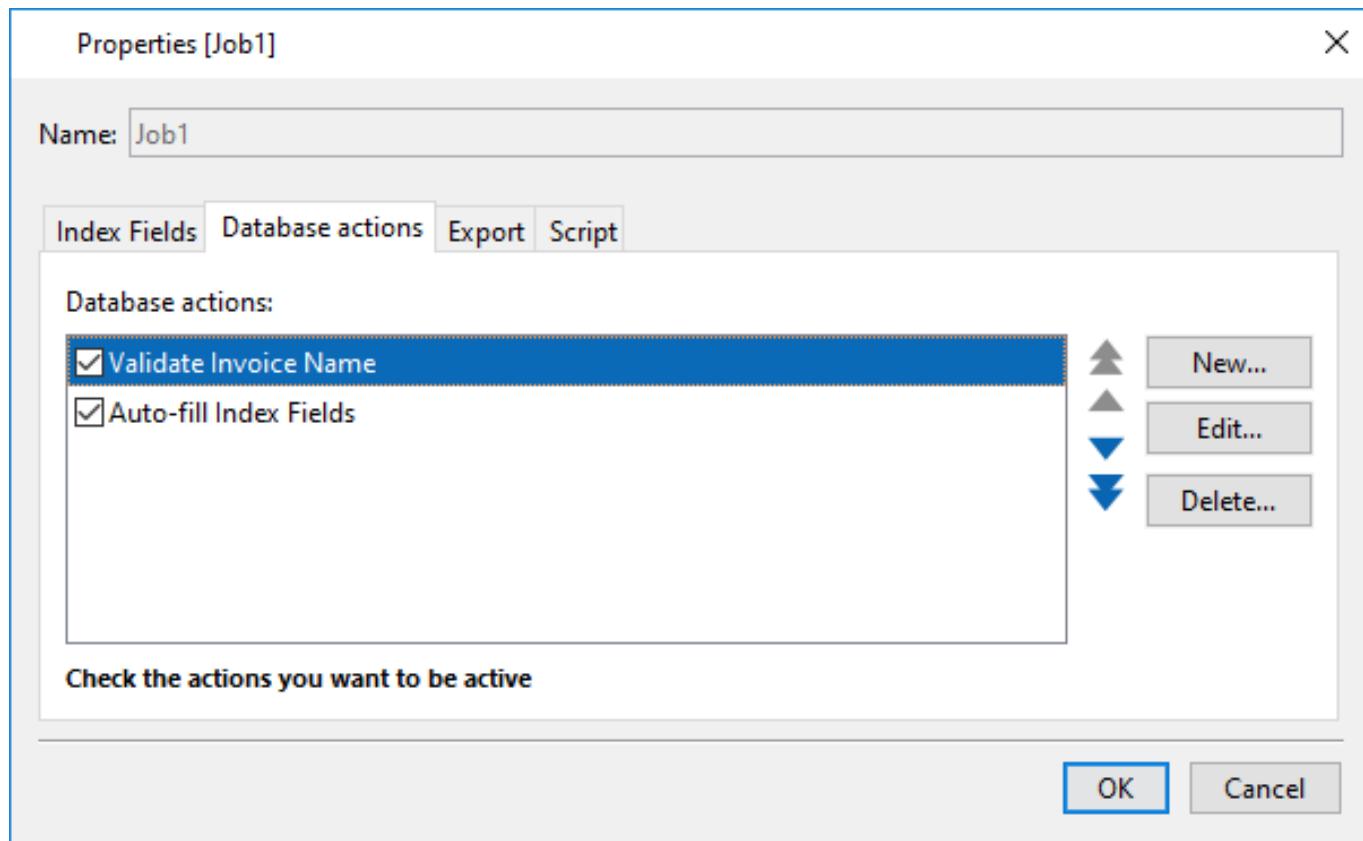


Figure 82. Database actions list

From the *Database Actions* list you can create, edit and delete *Database Actions*. *Database Actions* are executed in the order they are defined: you may change their order by using the *Top*, *Bottom*, *Up* and *Down* buttons. To disable a *Database Action* click the checkbox from the above list to uncheck it: you can thus disable it without deleting its definition.

Creating/Editing Database Actions

The *Database Action* dialog is used to create/edit a *Database Action*:

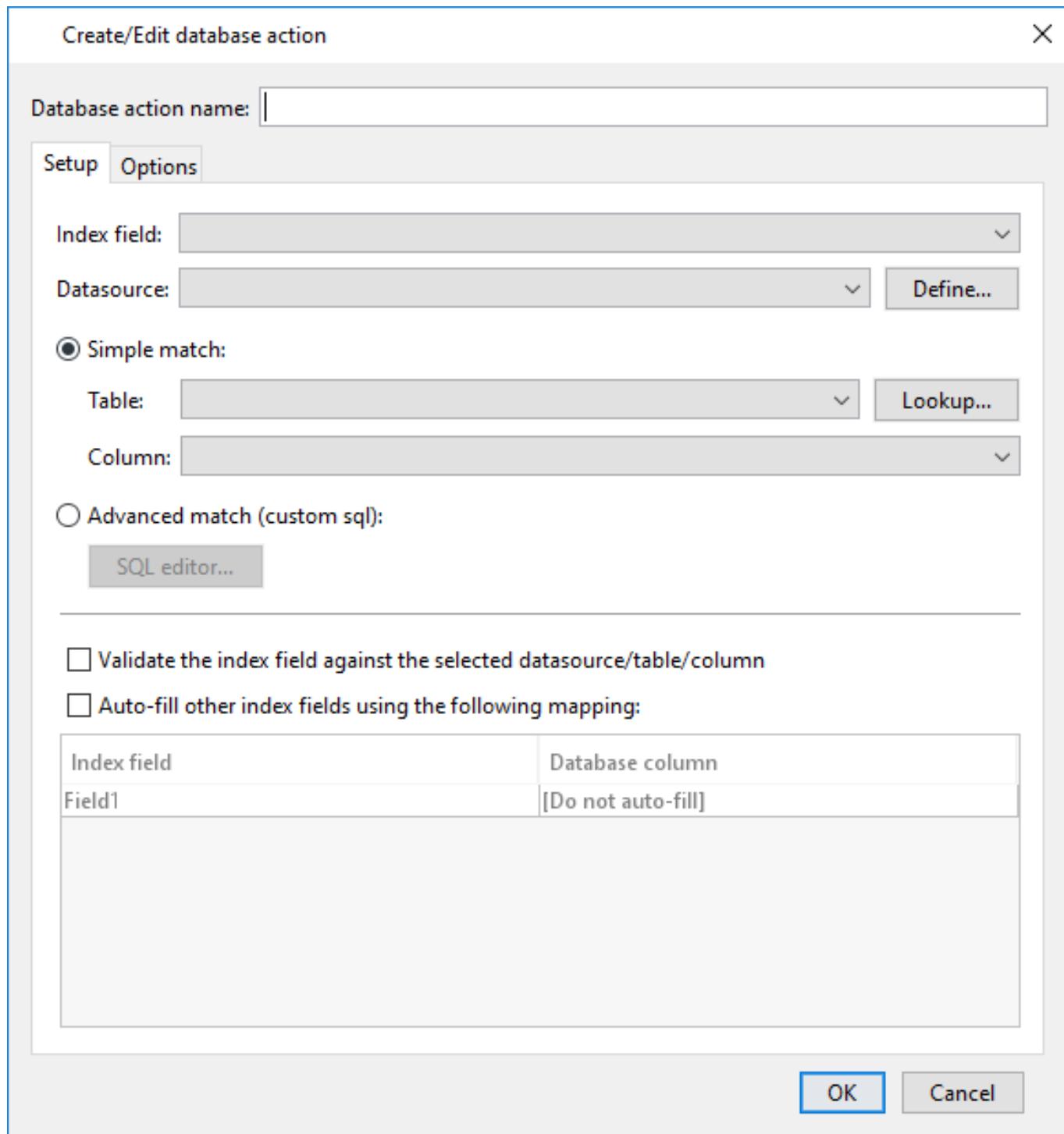


Figure 83. Database Action dialog: Setup tab

In the *Setup* tab, you must select the *Index Field* for which this *Database Action* will work. All available *Index Fields* will be shown in the *Index Field* combo box. Note that because a *Database Action* can be related to a *Document Class*, a *Folder Class* or *Batch-level indexing*, only the corresponding *Index Fields* of this *Document Class*, *Folder Class* or *Job* will be shown.

You need to define a connection to a database and you can do so by selecting the appropriate **Data-source** from the *Datasource* combo box.

Setting up the database query for validation

Once you have setup a *Datasource*, you have two options on how to perform a database action:

- *Simple match*: select the table and column against which the *Index Field* value will be validated
- *Advanced match*: manually write a SQL command (one or more queries, etc) to execute

Simple match

This is the easiest, and most common, way to perform a database validation: you just need to select the table and column from the *Table* and *Column lists*. If the table you want to use does not appear in the list, click on the *Lookup...* button on the right of the *Table list* to explore secondary catalog/schemas of the database.

Advanced match (custom sql)

With the *Advanced match* option, you can manually write the query that you want to be executed in order to validate the *Index Field*. Click on the *SQL editor...* button to open the *SQL Editor dialog*. The *advanced match* function allows you to:

- perform queries that make use of the values of any *Index Field* in their **where** clause
- perform more than one queries (use different queries to validate an *Index Field* value)
- write complex sql that is database specific

The following rules apply for the SQL you write:

- You can write one or more valid SQL statements separated with a semi-colon (;)
- If you need more than one select SQL statements, make sure that they return the same number of columns
- You can use column aliases to name your columns; in case of multiple select statements, the labels of the first statement will be used to populate the *Auto-fill list* for *Index Field mapping (Create/Edit database action dialog)*
- All statements you write will be executed in the same database transaction (for example you may write an *update/insert* statement first and use a secondary *select* statement to read the value you inserted/updated).
- You can use the *Index Fields* names as named parameters by prepending the colon before the *Index Field* name (e.g. `:firstname`)
- If you use named parameters, their data-types will be used for the SQL binding.

For example, let's assume you have an *Index Field* named **AccountID** and a second one named **Account-**

Type that you use to lookup an account in a remote database, and you want to get the *Customer Name* and *Balance* to auto-fill two other Index Fields. An advanced SQL to do that may look like that:

```
select CustName, Balance from Accounts where ID = :AccountID and AccType = :AccountType
```

In the above sample statement, the *CustName*, *Balance*, *ID*, *AccType* are column names of the *Accounts* table in the hypothetical database.

Click on the *Validate...* button in order to validate the SQL query: *Core Service* will execute the query against the remote database and also check the validity of any named parameters. Click *OK* to save and close the *SQL editor dialog*.

Auto-filling Index Field values

You can use the results of the database query (either simple or advanced) to validate the Index Field, but also to auto-fill the values of other *Index Fields*. To do so, check the *Auto-fill other index fields using the following mapping* option. The corresponding table will be filled with all available *Index Fields* on the left: click on the cell on the right column to select the database column that you want to use to auto-fill this field.

If you have selected *Advanced match*, then the custom SQL query will be executed against the database in order to retrieve the actual database columns to map to *Index Fields* at this step: this is why it is important to have a valid SQL query that validates in the previous step.

Selecting when a Database Action is triggered

Database Actions are executed when:

- an *Index Field* gets/loses focus, or
- when indexing of the current node (Batch, Folder, Document) begins/ends.

Properties [Contract] X

Name:

Description:

[Index Fields](#) [Database actions](#) [Export](#) [Script](#) [Form Types](#)

Database actions:

New...
Edit...
Delete...

Check the actions you want to be active

Save Cancel

Figure 84. Define Database Actions

You can select when the *Database Action* is triggered from the *Options* tab of the *Create/Edit database action dialog*. You can also select whether you want the *Database Action* to be executed before or after a script that may exist on the field/node level, from the *In relation to script* option.

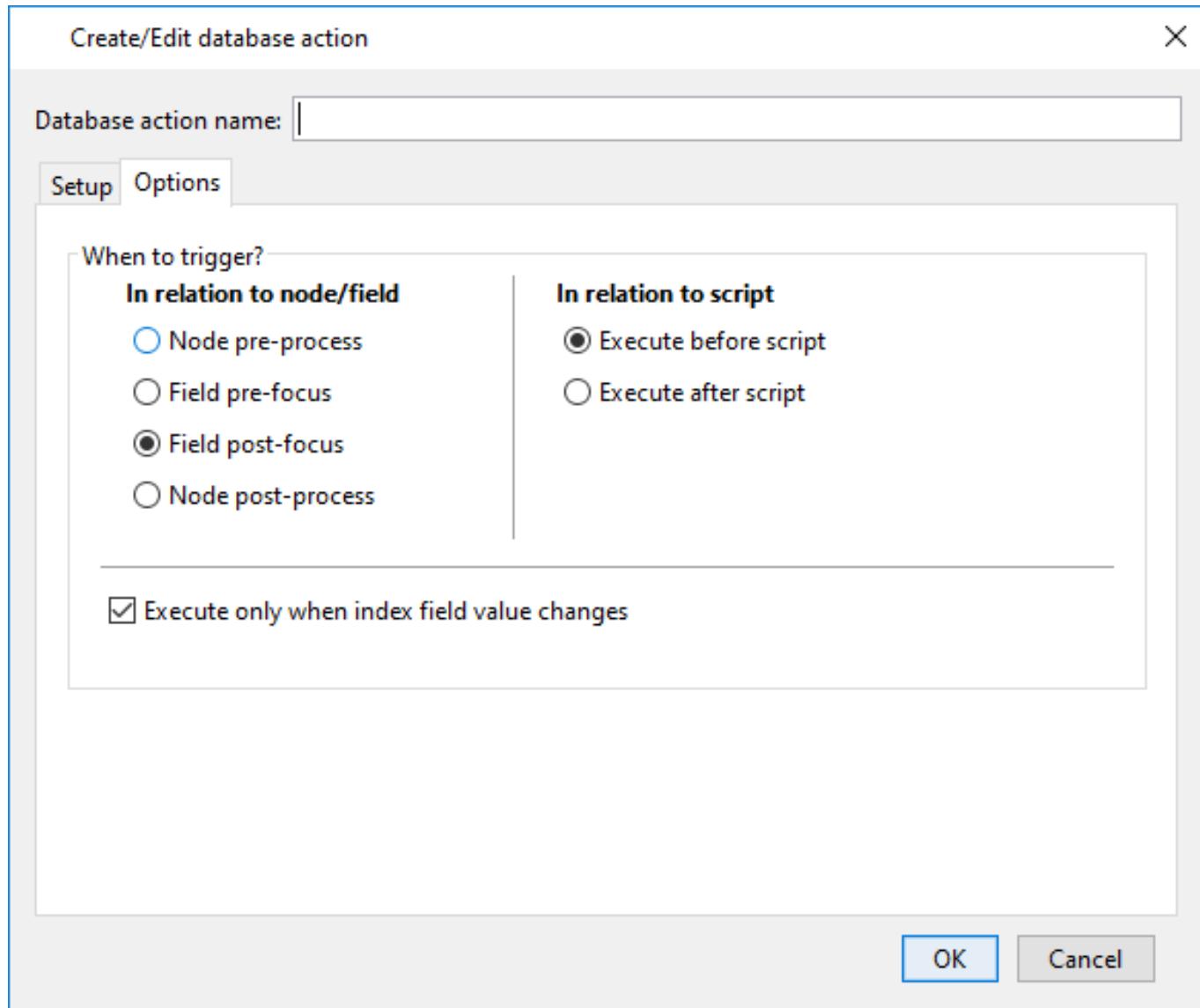


Figure 85. Database Action dialog: Options tab

By default, field-bound *Database Actions* are executed only when the value of the *Index Field* changes. If you want to execute the *Database Action* every time the focus enters/leaves the *Index Field*, uncheck the *Execute only when index field value changes* option in the *Options* tab.

3.1.12.5. Indexing Scripting

Info Input Solution allows you to fine-tune several aspects of the system by writing custom scripts using Javascript. You may attach a script that executes during indexing to perform custom validations and extend the functionality of *Info Input Solution*. *Indexing scripts* can be defined for [Batch level indexing](#), [Document Classes](#) and [Folder Classes](#) by clicking the *Script...* button:

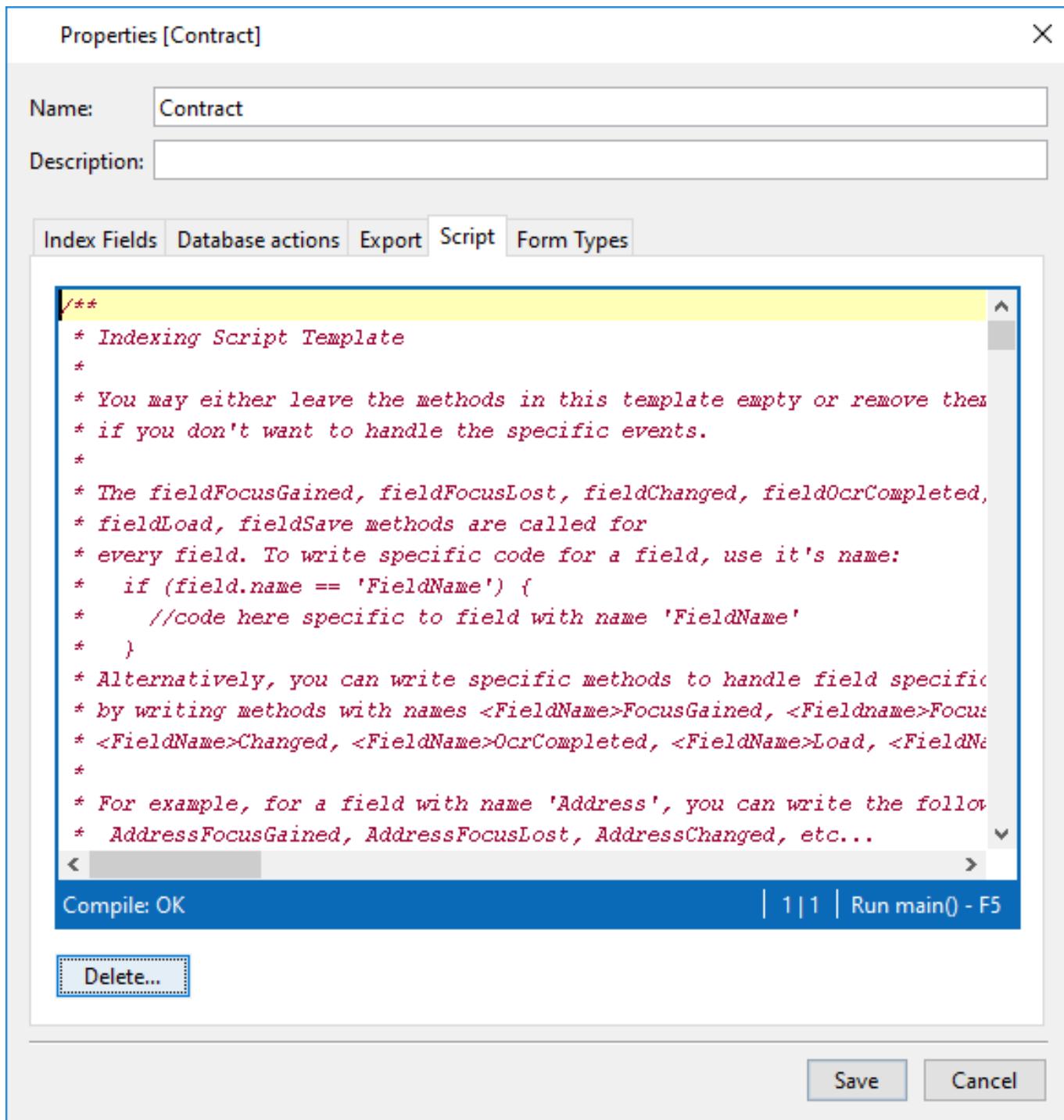


Figure 86. Javascript editor dialog

The first time you click the *Script...* button a default script is created and presented in the Javascript editor dialog. Empty function placeholders are pre-created in the default script so you can just fill-in the body of the ones you want to customize. The default script contains several comments at the top and before each function that help you understand what each function stands for. Please refer to the *Info Input Solution Developers Guide* for more details.

You can debug your *Indexing Scripts* using [test batches](#).

3.1.12.6. Advanced Indexing Forms

A custom *Indexing Form* can be designed at *Batch*, *Folder* or *Document* level, using the Advanced Indexing Form Designer GUI. These advanced Indexing Forms will be displayed instead of the default Indexing panel, when the batch is opened in Indexing mode, using the *HTML Client*.

Note: The Advanced Indexing Forms feature is not available for the *Thick Client*.

Advanced Indexing Form Designer

Advanced Indexing Form Designer is a graphical interface for creating and managing forms. It features a sidebar with various tools and a main workspace for form components.

Top Bar:

- Eye icon: Preview or visibility settings.
- CSS icon: CSS editor.
- JS icon: JavaScript editor.
- Pen icon: General editing or styling.

Indexing Fields:

- Field1:** Contains icons for Text (T), Date (D), and AB (AB).
- Field2:** Contains icons for Text (T), Date (D), and AB (AB).
- Field3:** Contains icons for Date (D), AB (AB), and AB (AB).
- Zone zoom:** Contains an icon for AB (AB).

Components:

- Text (T)
- Date (D)
- AB (AB)
- Checkmark (✓)
- Text (Abc)
- Image (Image)

Containers:

- Form
- Table
- List
- Group

Form Explorer:

- Form

The Advanced Indexing Form Designer must be used after the setup of the Indexing Fields for each Batch, Folder or Document Class level. When configured at Document Class level, a custom Indexing Form can be designed as common for the Document Class or multiple different custom Indexing Forms can be designed for each Indexing Form Type of the same Document Class.

The Indexing Form Designer offers a toolbar with the list of predefined Indexing Fields, a list of additional Components and Containers that can be used to add content to the Indexing Form.

Create a Default Layout Form

As a starting point, the Job administrator can use the Generate Default Layout wizard button to automatically add all configured Indexing Fields to the working Indexing Form, with or without Indexing Zone View elements next to each Field.

Adding custom JavaScript and CSS script code to the Indexing Form

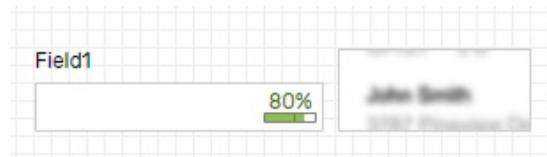
In addition to the designer tools that are available with the Advanced Indexing Form Designer GUI, it is also possible to add custom scripting code for the Indexing Form and its elements. A JavaScript Form Editor and CSS Editor window is provided. The JavaScript Form Editor also provides access to the Batch, Folder or Document Class level Indexing script.

Indexing form Preview

When designing a custom Indexing Form, the *Preview* button will show a pop-up window with the Form Preview. Note: The CSS script code will be also executed in Form Preview mode, but the JavaScript Form code is only partially functional in Form Preview, because a long list of batch properties, variables and functions that are available at runtime in the *HTML Client* are not available in Form Preview mode.

Indexing Fields

Every Indexing Field can be displayed with three components: The Field box (with optional Confidence value indicator from an Extraction processing step), the Field label (Field description) and the Field Zone View. These components can be added as combo to the Indexing Form, or each one of these elements can be drag-n-dropped separately.



A Zone zoom element is also available at the bottom of the Indexing Fields list. This element will dynamically change and show the Zone View of the Indexing Field with cursor focus.

Components

The following components can be added as elements in the Indexing Form. The General properties (Name, Classes) can be defined and later referenced in CSS and JavaScript Form scripts. A long list of CSS properties can be further configured in the *Layout, Display, Presentation, Content* and *Text Properties* sections. Additional CSS style rules can be defined in CSS script code.

- Label
- Text box
- Radio button
- Check box
- Button
- Image

Containers

The following list of Container options can be combined to setup an Indexing Form with sections that can be easily browsed by the Indexing operator, for better organization of a long list of Indexing fields in smaller sections. The viewing sequence of these Containers can be further customized with JavaScript Form scripting and Indexing scripting hook functions.

- Panel
- Titled Panel
- Tab view
- Columns view

Form Explorer

The structure of the *Indexing Form* with *Containers* and *Components* in each Container is dynamically updated and displayed in the Form Explorer.

Tab Order

The operation of the Tab key can be customized and the Tab Order for the full list of the Index Field boxes, custom Text boxes, buttons, checkbox buttons and radio buttons can be modified in the Tab Order panel.

3.1.12.7. Localization

Date, time and numbering format follow the user-locale specific settings, both in the *Indexing* pane and in the ranges definition in [FieldTypeDialog](#). The actual values are always stored in the Database using the 'standard' notation (e.g. dot '.' for decimal, comma ',' for thousands separator). *Indexing* scripts will use the locale-specific settings when accessing such fields, whereas all other scripts will use the stan-

dard.

Locale specific scripting behavior is further explained in the *Info Input Solution Developer's Guide*.

3.1.12.8. Rubber band OCR

During Indexing, the rubber band Optical Character Recognition (OCR) functionality can be used to recognize and extract field values from the currently processed image file, and auto-fill the values of *Index Fields*.

Supporting multiple languages

Multiple languages can be supported for the OCR functionality. Info Input Solution is shipped with the following language files, under `add-ons\ocr-data\`.

Language name	Language code	enabled
English	eng	yes
Arabic	ara	no
Chinese Simplified	chi_sim	no
Chinese Traditional	chi_tra	no
French	fra	no
German	deu	no
Japanese	jpn	no
Polish	pol	no
Spanish	spa	no
Turkish	tur	no
MICR fonts	mcr	no

By default, only the English language OCR library is enabled during installation. To enable other language library, available on Info Input Solution, a system administrator should review the already available language files, under `<installation path>\add-ons\ocr-data\`, and enable the desired language by removing the comment signs (#) to uncomment the corresponding section in the `index.txt` file. Any modification will have immediate effect on new *Client* logins. If the same *Extraction profile* will be used in an Extraction or Intelligent OCR step, a Core Service restart will also be required for the new languages to be loaded

MICR Fonts

The MICR fonts language pack is used to recognize MICR codes from scanned cheques. The banking industry uses Magnetic Ink Character Recognition Code (MICR Code) fonts for processing cheques and other documents. The MICR language file provided is for the MICR E-13B font. Besides decimal digits, it also contains the following symbols:

- transit: used to delimit a bank branch routing transit number
- amount: used to delimit a transaction amount
- on-us/account: used to delimit a customer account number
- dash: used to delimit parts of numbers - e.g. routing numbers or account numbers.

The OCR engine converts the special characters according to the following table:

Character	OCR output char
transit	[
amount	@
on-us / account	-
dash	\$

Additional languages

Additional OCR language files can be downloaded from the tesseract-OCR repository, available at <http://imageaccesscorp.com/Files/Support/IT/Tesseract-OCR-language-files> and added to the destination path mentioned earlier. For every new OCR language file, a new unique section is required in `index.txt`. The allowed properties are: `locale`, `file`, `name` and `nativeName`. The `locale` is the Java language locale, using 2-character `ISO 639-1` language codes. The `file` name format (`tesseract-ocr-x.xx.lng.tar.gz`) uses 3-character `ISO 639-2` language codes.

As of June 2016, there are language packs available for the following languages: ara (Arabic), aze (Azerbaijani), bul (Bulgarian), cat (Catalan), ces (Czech), chi_sim (Simplified Chinese), chi_tra (Traditional Chinese), chr (Cherokee), dan (Danish), dan-frak (Danish (Fraktur)), deu (German), ell (Greek), eng (English), enm (Old English), epo (Esperanto), est (Estonian), fin (Finnish), fra (French), frm (Old French), glg (Galician), heb (Hebrew), hin (Hindi), hrv (Croatian), hun (Hungarian), ind (Indonesian), ita (Italian), jpn (Japanese), kor (Korean), lav (Latvian), lit (Lithuanian), nld (Dutch), nor (Norwegian), pol (Polish), por (Portuguese), ron (Romanian), rus (Russian), slk (Slovakian), slv (Slovenian), sqi (Albanian), spa (Spanish), srp (Serbian), swe (Swedish), tam (Tamil), tel (Telugu), tgl (Tagalog), tha (Thai), tur (Turkish), ukr (Ukrainian), vie (Vietnamese).

3.1.12.9. Annotations

The *Annotations* module can add custom shapes, comments, text-boxes, highlighted areas or rubber stamps anywhere on the image. The module is available in *Single Page Viewer* and *Indexing Pane*.



Annotations are used as metadata information, attached to the image. The following two buttons are used for viewing and editing the Annotation properties.

image:add_annotations.png[pdfwidth=100%,align="center"]	Show/Hide Annotations will make all currently added Annotations visible (or invisible) in <i>Single Page Viewer</i> or during <i>Indexing</i> , without deleting them from the current batch.
	Edit Annotations enables the <i>Annotations toolbar</i> for adding new Annotations and / or modifying the existing ones.



Figure 87. Annotations toolbar

3.1.12.10. Annotations toolbar

Default Rubber Stamp Annotations

The rubber stamp icon (image:rubberStamp.png[pdfwidth=100%,align="center"]) offers a set of text rubber stamp Annotations. The default text list is: APPROVED, AUTHORIZED, BANNED, CENSORED, DENIED, OK, PASS, QUALITY, REJECTED, SOLD. This list of rubber stamps is also available (translated) in all supported [User Interface languages](#). Note that, once a rubber stamp is applied to an image, it will not be translated, if the display language is changed.

An administrator can replace this list with custom ones, possible different for every User Interface language. An admin can also add custom images to the system, either language-specific or not.

Updating the set of text Rubber Stamps

An admin can overwrite and define one or multiple custom lists of text rubber stamp Annotations. These custom lists can be language-specific or universal, but will not be automatically translated. For a language-specific list of rubber stamps, a new property `rubberstamps_text_LC` must be defined, where

LC: 2-character ISO 639-1 language code. Similarly, the property `rubberstamps_text` will define a new universal list of rubber stamps. The Thick Client will first check if a language-specific `rubberstamps_text_LC` property exists, matching the current User Interface language. If not, it will look for a universal property `rubberstamps_text` with custom rubber stamps. Otherwise, the default text list will be used.

For example, defining the property `rubberstamps_text_en=APPROVED|AUTHORISED|BANNED` will offer this new list of text rubber stamps in the English language UI, instead of the system default list. When using the Thick Client in other languages, the default list will be available. More details on defining Core Service Parameters are available in [Server Configuration Parameters](#) section.

Adding Image Rubber Stamps

A set of custom images can also be added as rubber stamp Annotations. An admin must create an archive file (`.jar` and/or `.zip`) with custom images. The archive can optionally have folders for language-specific rubber stamps (2-character ISO 639-1 language code, e.g. `/es/` and `/ar/`) and may also contain a default set of images in `/` as default fallback. The Thick Client will check for images in the corresponding language sub-folder of the archive, and will add them to the rubber stamp toolbar. If a language sub-folder is not found, then the default fallback images will be added, if any. Note that, these Image Annotations will not overwrite the rubber stamp text list.

This image archive must reside on the Core Service file system, below the `client` folder. The Core Service property `rubberstamps_path` must be defined with a relative path (including the archive file name, e.g. `resources/rubberstamps.jar`), which is resolved at runtime using the applet's codebase. A good practice would be to put the archive file alongside the applet's jar files, or in a sub-folder therein. Also note that, the *Export Service* procedure must have access to this archive file, for burning the annotations on the image during Export. The corresponding property `released.page.image.directory=<archive_path>` must be set in the `released.properties` file, located under `<installation path>\released\`.

bUser-defined Image Rubber Stamps

A user can also add and use a custom user-defined Annotation image, by scrolling down the *Rubber Stamp Properties* list window.

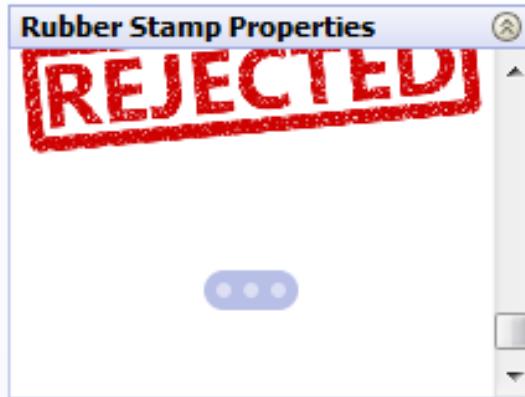


Figure 88. Add a user-defined Annotation image

3.1.12.11. Add a user-defined Annotation image

The custom image will be uploaded to the Core Service, along with the batch. Adding more than one custom image is possible, and all of them will be uploaded for further processing. However, only the latest user-defined Annotation will be available in the *Annotations toolbar*.

3.1.13. Release

Info Input Solution enables data export through the export subsystem to allow reuse of the scanned images across applications. The export subsystem was designed to be fully configurable and easily extensible, from the user interface down to the *Export Service*.

Data *Export* is the last step in the Info Input Solution process. Captured data (page images and / or indexed data) is exported to some external system for further processing or archive.

The export step is automated and runs in the background. The administrator sets up the *Export Destinations* on the system and uses them to create custom *Export Configurations* per *Job*.

3.1.13.1. Export Service

The *Export Service* is a separate process that runs independently of the Core Service, as a separate OS process. The *Export Service* is a multi-threaded, high-throughput process that can be installed on the same physical machine as the Core Service, or on separate machine(s). The *Export Service* needs to have access to the directory where the Core Service stores the batch images, so if an *Export Service* instance is installed on a different physical machine, it still needs to have network access to the common storage system.

The first time an *Export Service* connects to an Core Service, it registers itself as an active *Export Service* instance that participates in the *group of Export Services* that may also run against the same Info Input Solution installation. You may see a list of all registered *Export Service instances* for your Info Input Solu-

tion installation from the *Tools & Options* menu → *Export Server administration...* item:

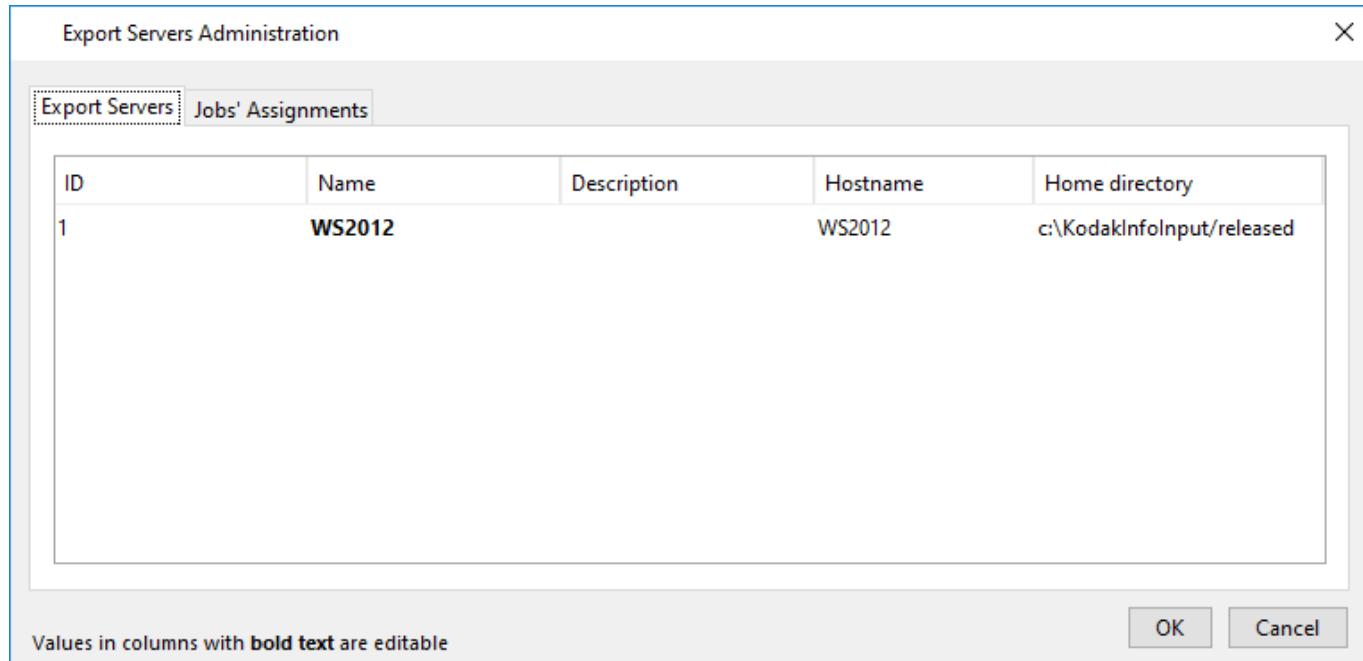


Figure 89. Export Server Administration dialog

If there are more than one *Export Service* instances that process *Batches*, then it is possible to assign a *Export Service* to only handle the export of batches belonging to specific *Job(s)*. You can do this from the *Jobs' Assignments* tab:

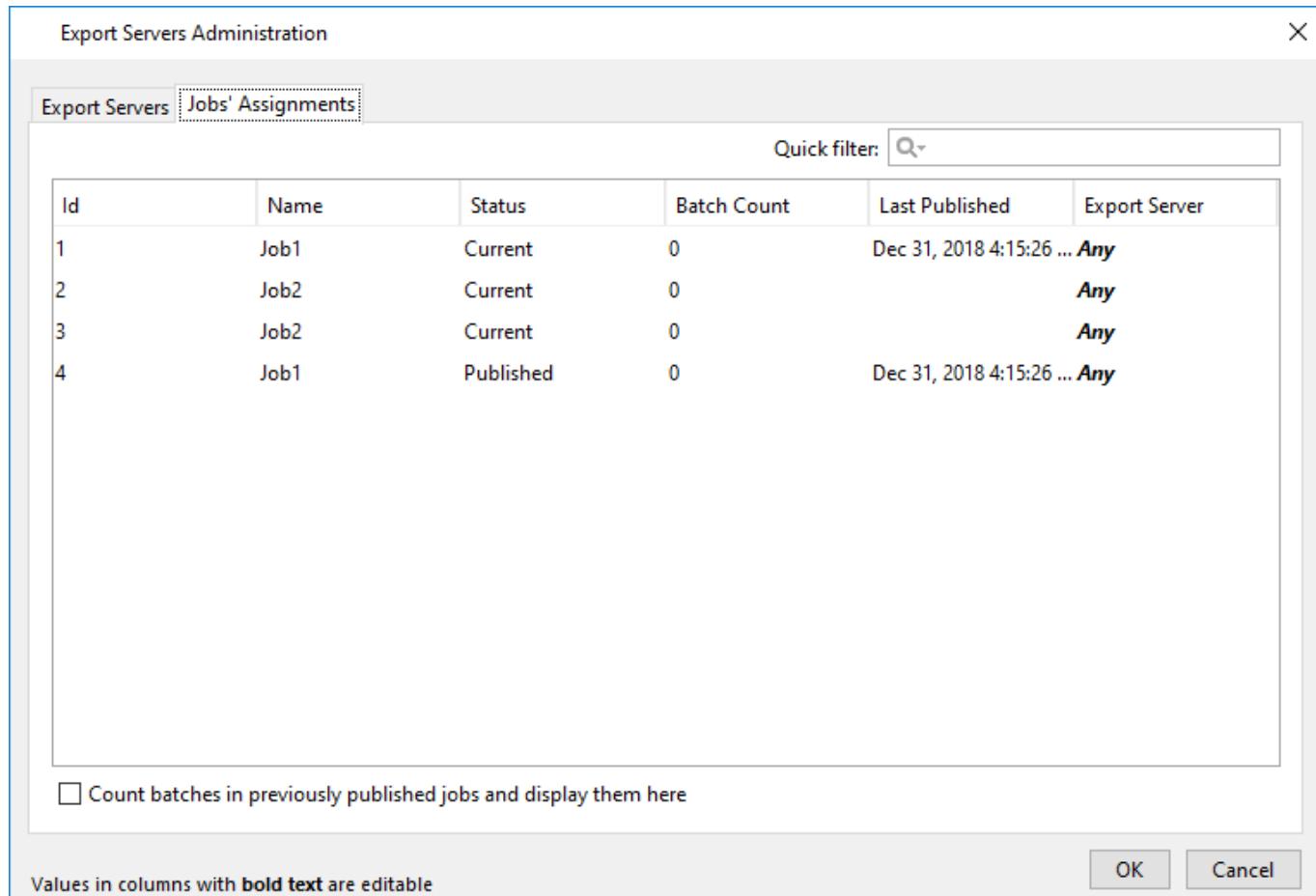


Figure 90. Export Server Administration dialog: Jobs' Assignments

Clicking on the *Export Server* column, you can select which *Export Service instance* you want to assign to each *Job* in the system. If you choose the value "Any", then this *Export Service* will pick batches from any *Job* to process.

Notice that multiple lines may appear in the above table for the same *Job*: there is a single line for each time you have published the *Job*, along with the *Last Published* column. If you check the *Count batches in previously published jobs and display them here* option, then the *Batch Count* column is populated to help you see how many batches exist in the *Export Queue* - beware that this is an option that can take time.

There is always a line for each *Job* where the *Status* is *Current*: this represents the *unpublished Job* (or the *Job* at its current state in the [Jobs tab of the Setup data dialog](#)): if you set the *Export Service* column for this line, then whenever you publish this *Job*, the new *Job copy* that is created will inherit this value.

3.1.13.2. Export Configurations: How the export of Batches, Folders and Documents works

Info Input Solution allows the export of data to be independently configured on the *Batch*, *Folder* and *Document* level.

You can create/edit *Export Configurations* from the *Export Configurations* dialog which is accessible from several paths:

- Batch level export: *Setup data dialog* → *Jobs tab* → *Job Properties dialog* → *Indexing & Export Tab* → *Export...*
- Document Class export: *Setup data dialog* → *Document Classes tab* → *Export...*
- Folder Class export: *Setup data dialog* → *Folder Classes tab* → *Export...*

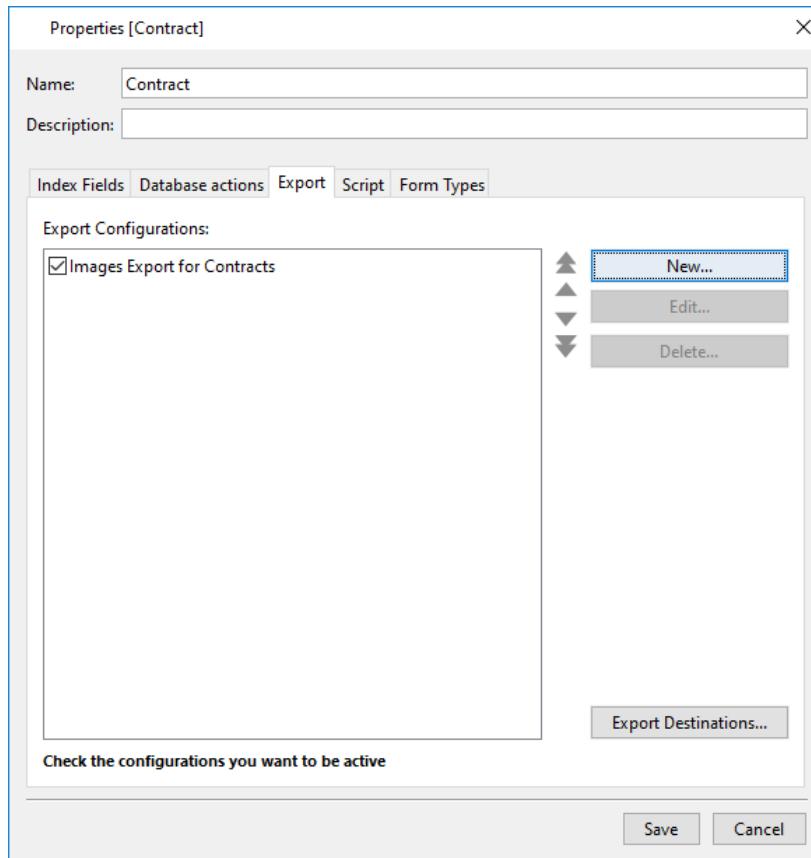


Figure 91. Export Configurations dialog

An *Export Configuration* is an in-process module that exports *Batch* data (images, meta-data or both) to an external system. You can setup any number of *Export Configurations* for the *Batch level* and for any *Folder* and/or *Document Class*. *Export Configurations* are executed for each node in the order they have been defined. If you have defined different *Export Configurations* for different levels (e.g. for the *Document* and *Folder* level), then these will be executed in parallel.

Each *Batch* is guaranteed to be exported by a single *Export Service*, although multiple *Export Services* may export *Batches* belonging to the same *Job*: this peculiarity leaves no guarantees on the order that *Batches* belonging to the same *Job* will be exported, in case there are multiple *Export Services* that have been assigned to export data from the same *Job*.

Every *Export Configuration* is based on a *Export Destination*. *Export Destinations* are *Shared* objects which can be used by multiple *Jobs*, *Document Classes* or *Folder Classes*. An *Export Configuration* is a customized version of an *Export Destination* whose parameters are modified to apply to the type of data being exported. You can think of an *Export Destination* as a system resource and an *Export Configuration* as the customized version of this resource. For example, the *Disk Export Destination* is a module that exports data into disk files. The *Disk Export Destination* has certain parameters that define its behaviour, like the output folder. An *Export Configuration* for a *Document Class* can use the *Disk Export Destination*, but define its own specific parameters for exporting the *Document* data. For example it could define a specific filename naming format so that data belonging to this *Document Class* are easily identified.

The exact context of a *Folder* or *Batch* level *Export Configuration* depends on the implementation of the *Export Destination* (even the support of more than one levels). For example, the *Images Export Destination*, that exports images on the file system, saves the images of documents, when it is configured on the *Document* level, but it is not really obvious what it shall do when it is configured on the *Folder* level, since *Folders* do not have associated images: in this specific case, the *Images Export Destination* outputs the images of all the documents that the folder contains, when configured on the *Folder* level. Yet, this behavior is not really obvious and depends on the implementation of each *Export Destination* module.

Info Input Solution supports the custom development of *Export Destination* modules using Java that allows you to export images and meta-data to legacy or proprietary systems, or by using custom web-services. Please refer to the *Info Input Solution Developer's guide*, under section "3.5. Export Destination plugin development" for more information.

3.1.13.3. Setting up Export Configurations

From within the *Export Configurations dialog*, click on the *New..* button. This will bring up a dialog where you must select which *Export Destination* to use:

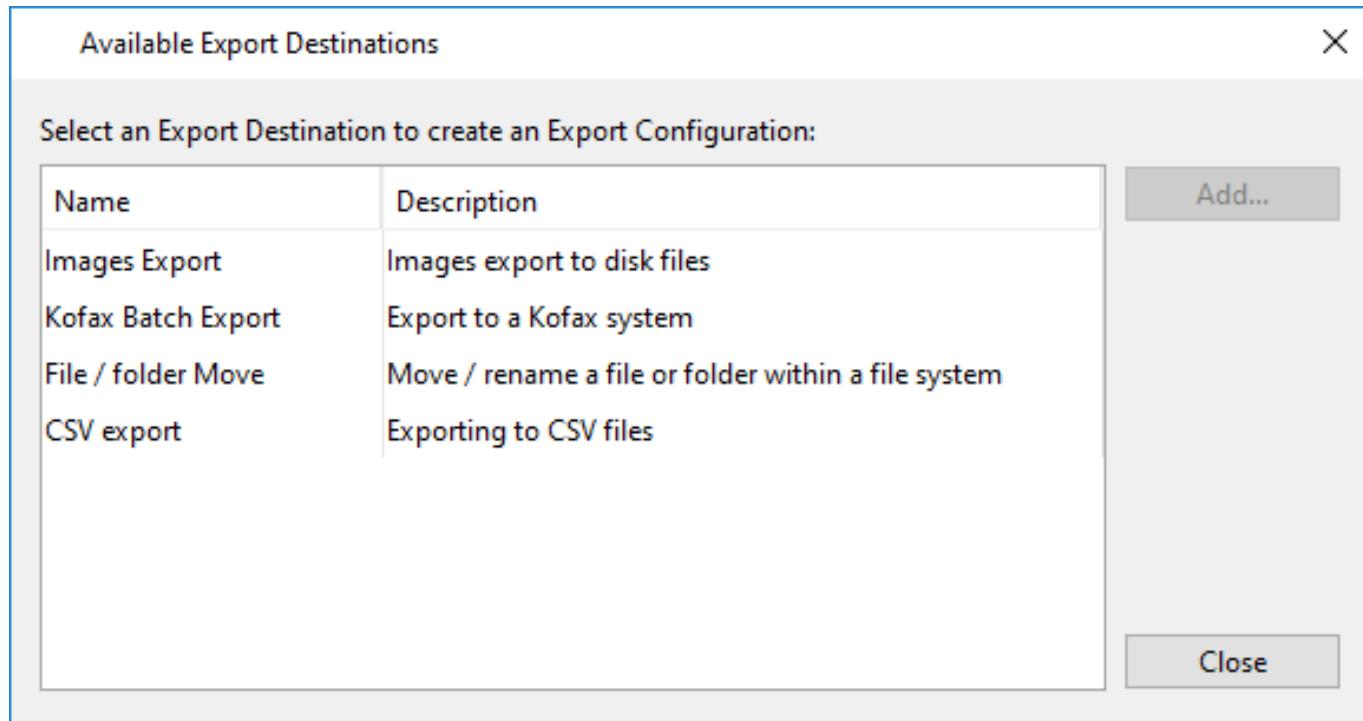


Figure 92. Export Destinations selection dialog

This dialog displays all *Export Destinations* available on the system. [Read here](#) more details on how to setup the system's Export Destinations.

Select the *Export Destination* you want to use and press *Add...* This will bring up the corresponding dialog for creating an *Export Configuration*:

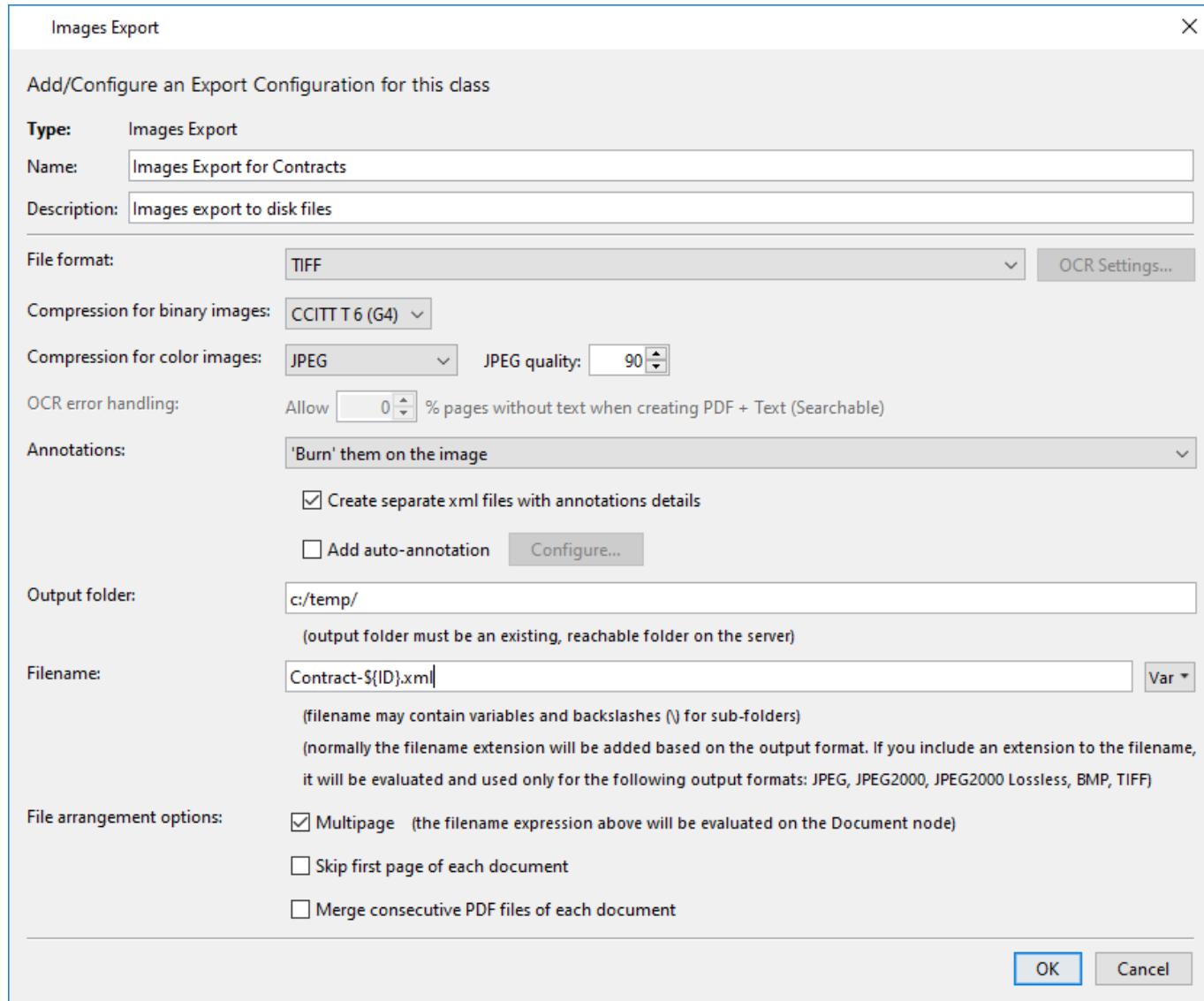


Figure 93. Export Configuration edit dialog

In this dialog, you may define the specific parameters for your *Export Configuration*.

3.1.14. Custom Workflow Actors

3.1.14.1. How Custom Workflow Actor works

Actors are the Info Input Solution's system components responsible for performing all processing work on Nodes (Batch, Folder, Document). Info Input Solution's workflow system is comprised of several such systemic actors that perform specific processing activities (e.g. batch creation, indexing, export, classification, extraction etc). Info Input Solution provides developers the ability to implement custom Actors in Java which can be placed anywhere in the workflow system and perform any custom logic on documents and batches.

Each Job in ImageTrust is associated with a Workflow, which is essentially a graph of Steps: each Step describes the execution of a logical operation which is performed by a registered/known Actor. When the ImageTrust Core Service starts it scans for custom Actor implementations, and registers them with the workflow system.



At runtime, if a Job uses a Workflow with a step that uses a registered custom Actor, the system will perform work using the custom Actor's processing logic.

For more information and an example on the configuration of Custom Workflow Actors please refer to the Developer's Guide under the corresponding section.

3.2. Shared Objects

Info Input Solution allows you to define certain objects, like *Document Classes*, *Field Types* or *Scan Profiles* and reuse them in multiple *Jobs*. These objects are called *Shared* and they are:

- [Document Classes](#)
- [Folder Classes](#)
- [Field Types](#)
- [Image Enhancement Profiles](#)
- [Extraction Profiles](#)
- [Data Sources](#)
- [Scan Profiles](#)
- [Export Destinations](#)

3.2.1. Document Classes

A *Document Class* is the definition of a specific type of document, and includes the specification of:

- a set of [Form Types](#)
- a set of [Index Fields](#), along with their types and specific properties
- one or more [Database Actions](#), that are executed when the value of an *Index Field* changes
- a *JavaScript script* where you can fine-tune the run-time behavior of the *Index Field* values
- a set of *Export* configurations for this type of document when processing ends

You can manage *Document Classes* from the *Setup Data dialog*, at the *Document Classes tab*:

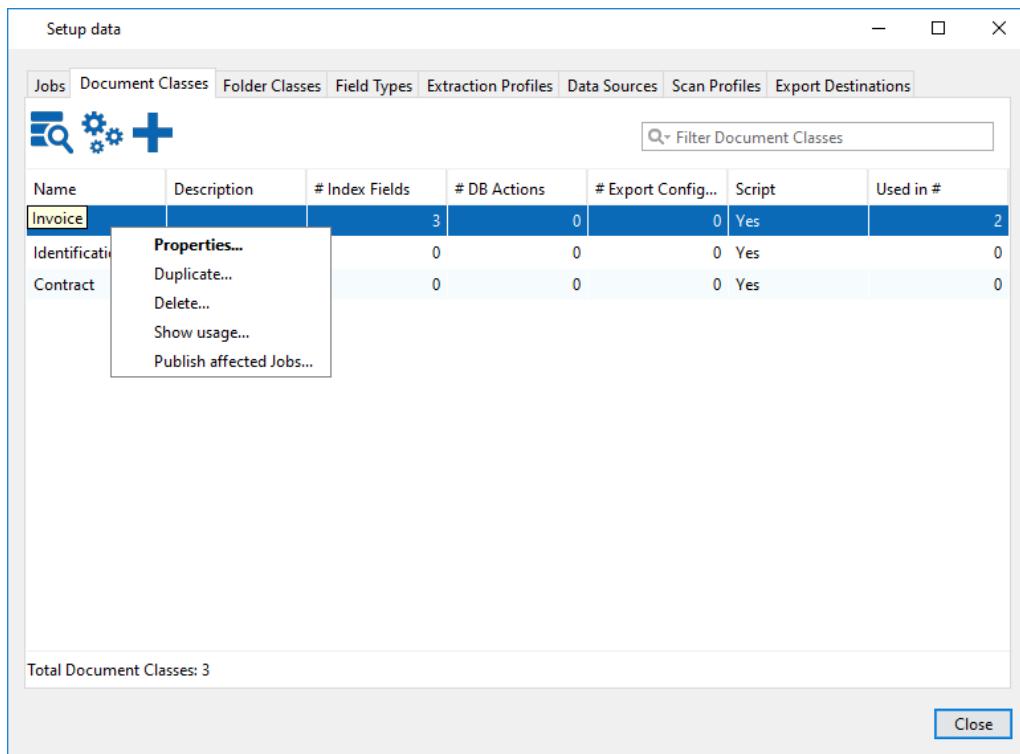


Figure 94. Setup data dialog: Document Classes tab

You may access the related functions from the popup menu by right-clicking on the rows in the table.

A *Document Class* needs to have one or more [Form Types](#). Each *Document Class* has a default *Form Type* created initially, which you can edit accordingly.

3.2.1.1. Assigning Document Classes to Jobs

Document Classes are *Shared* entities, which means that different *Jobs* may use the same *Document Class*. You associate a *Document Class* with a *Job* from the *Indexing & Export* tab of the *Job Properties* dialog:

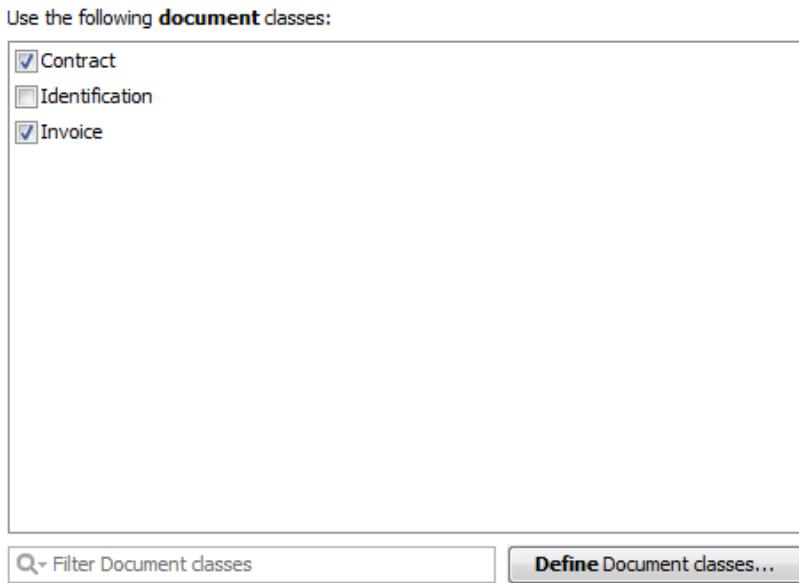


Figure 95. Associate Document Classes with a Job

This list displays all defined *Document Classes* in the system. You should check the ones you want to associate with the current *Job*.

During indexing, the index operator is actually choosing *Form Types* to associate with documents, not *Document Classes*. Since you, as an administrator, associate one or more *Document Classes* with a *Job*, Info Input Solution takes all the *Form Types* of all the *Document Classes* that are associated with a *Job* and presents them as a unified list to the index operator to choose from, during indexing. When the index operator selects the desired *Form Type*, the *Document Class* is indirectly being selected, the one where the *Form Type* belongs to, and thus Info Input Solution presents then the proper *Index Fields* to the index operator to proceed. Notice that all *Form Types* that belong to the same *Document Class* share the same index fields.

The following chart shows the association between a *Job*, *Document Classes* and *Form Types*:

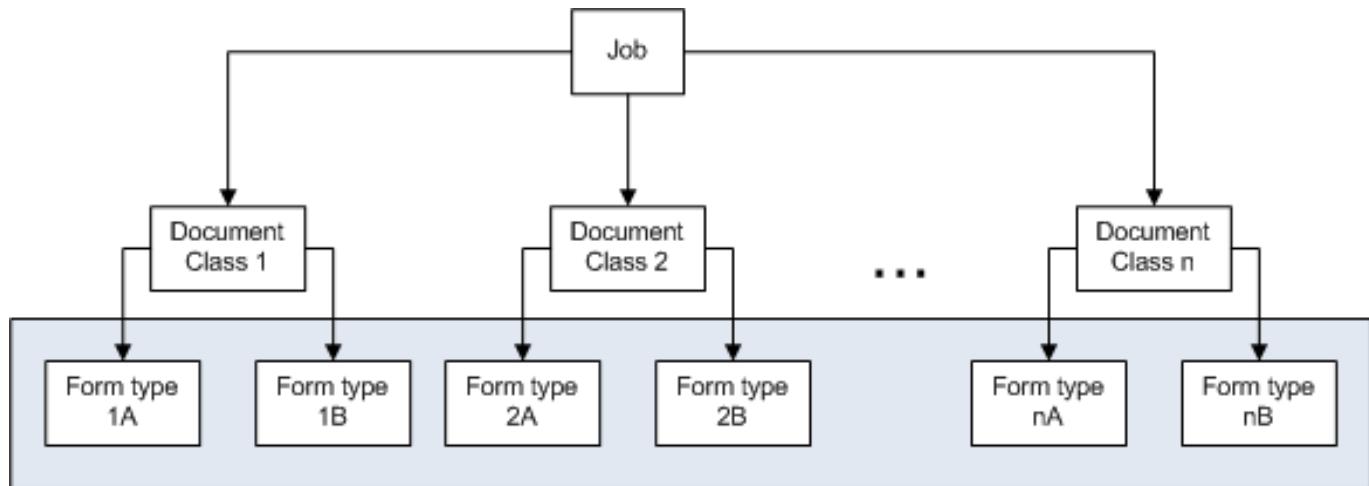


Figure 96. Job, Document Classes and Form Types

3.2.2. Folder Classes

A *Folder Class* is very similar to a [Document Class](#) with the exception that it is used to classify a *Folder* rather than a *Document*. The definition of a *Folder Class* includes the specification of:

- a set of [Index Fields](#), along with their types and specific properties
- one or more *Database Actions*, that are executed when the value of an *Index Field* changes
- a *JavaScript script* where you can fine-tune the run-time behavior of the *Index Field* values
- a set of *Export* configurations for this type of folder when processing ends

You can manage *Folder Classes* from the *Setup Data dialog*, at the *Folder Classes tab*:

Name	Description	# Index Fields	# DB Actions	# Export Config...	Script	Used in #
Contracts Folder		0	0	0	Yes	0
Invoices Folder		0	0	0	Yes	0

Figure 97. Setup data dialog: Folder Classes tab

All functions in this panel are identical with the ones in the *Document classes* panel, with the exception that a *Folder Class* does not have any associated *Form Types*.

3.2.2.1. Assigning Folder classes to Jobs

Folder Classes are also shared entities, like *Document Classes* and so they are shared amongst different *Jobs*. You associate *Folder Classes* with a *Job* from the *Indexing & Export* tab of the *Job Properties* dialog, by checking the ones you want to use in the *Folder Classes list*.

During indexing, when the index operator selects a *folder* to index, Info Input Solution presents a list of all *Folder Classes* associated with the *Job* to the user to choose from: when the index operator chooses one, Info Input Solution then displays the appropriate list of *Index Fields* of the selected *Folder Class*.

3.2.3. Field Types

A *Field Type* defines the type of data that can be stored in an *Index Field*. An *Index field* represents a piece of information (like a *name* or an *address* or a *date*) that comes from a document, or represents a property of the system (like the *scan date*), and can be exported at the end to a permanent storage. Apart from the actual type (e.g. whether it is a date or a number), a field type has a few more properties that may restrict its permitted values. Field Types are also used to define a specific list of allowed values for the user to select from (for example a list of Application Types, or a list of Colors like *Red*, *Blue*, *Green*, that will be displayed in a drop-down list for the user to pick one).

Field Types are *Shared* objects in Info Input Solution. Each *Index Field* in Info Input Solution must have a *Field Type*; if two *Index Fields* have the same *Field Type* and you change the properties of the *Field Type*, then both *Index Fields* that are linked to this *Field Type* will use the new properties.

3.2.3.1. Creating/Editing Field Types

Field Types can be managed from the *Field Types tab*, in the *Setup data dialog*:

Setup data

Jobs Document Classes Folder Classes **Field Types** Extraction Profiles Data Sources Scan Profiles Export Destinations

   Filter Field Types

Name	Description	Type	List	Used in #
Age	Integer number used for age	Integer	No	1
Color	List of colors	String	Yes	0
Date month/day/year	Date in month/day/year format	Date	No	0
String	Generic string with length 200	String	No	1

Total Field Types: 4

Close

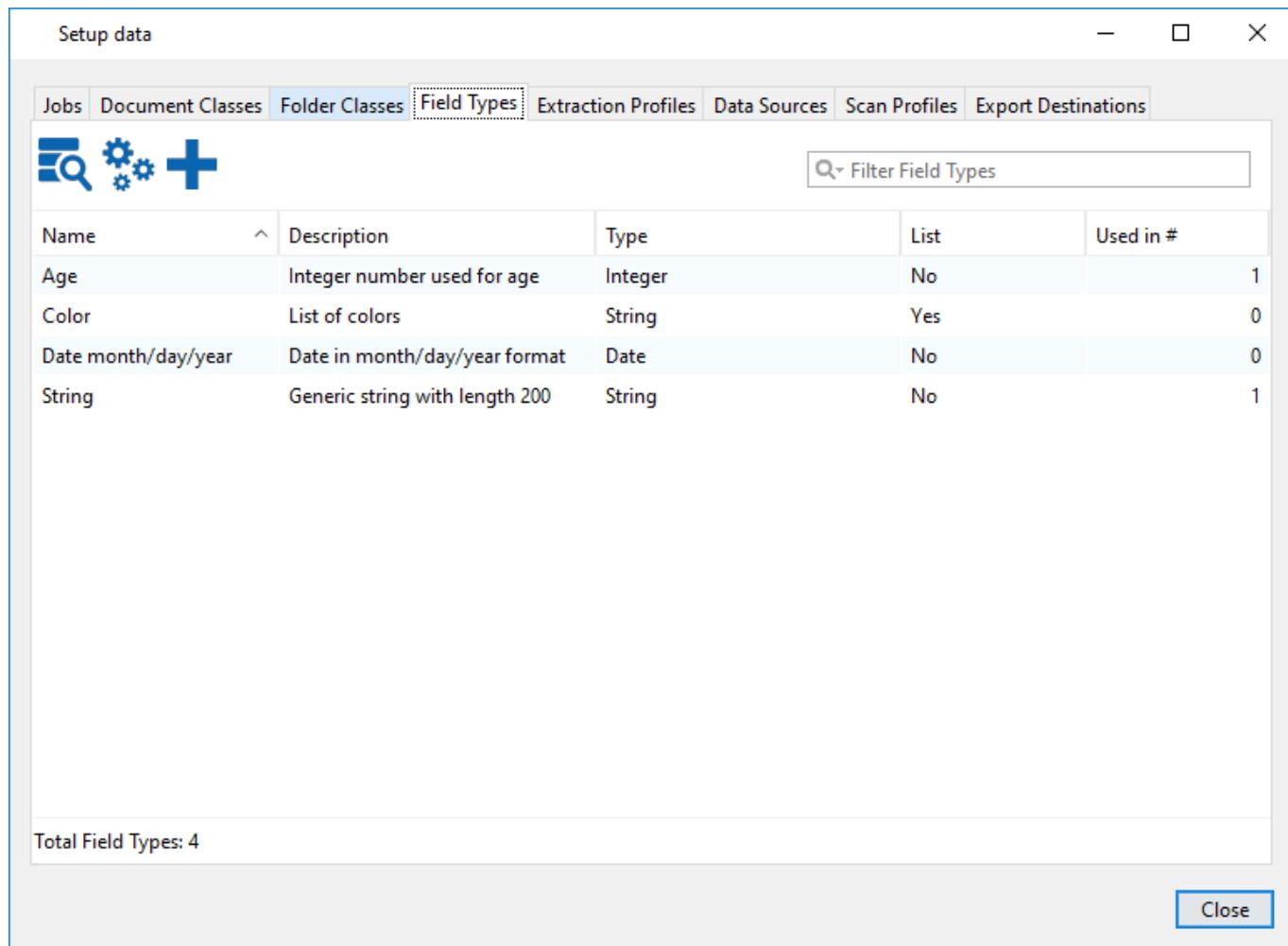


Figure 98. Setup data dialog: Field Types tab

The *Field Type Properties dialog* allows you to create/edit a *Field Type*:

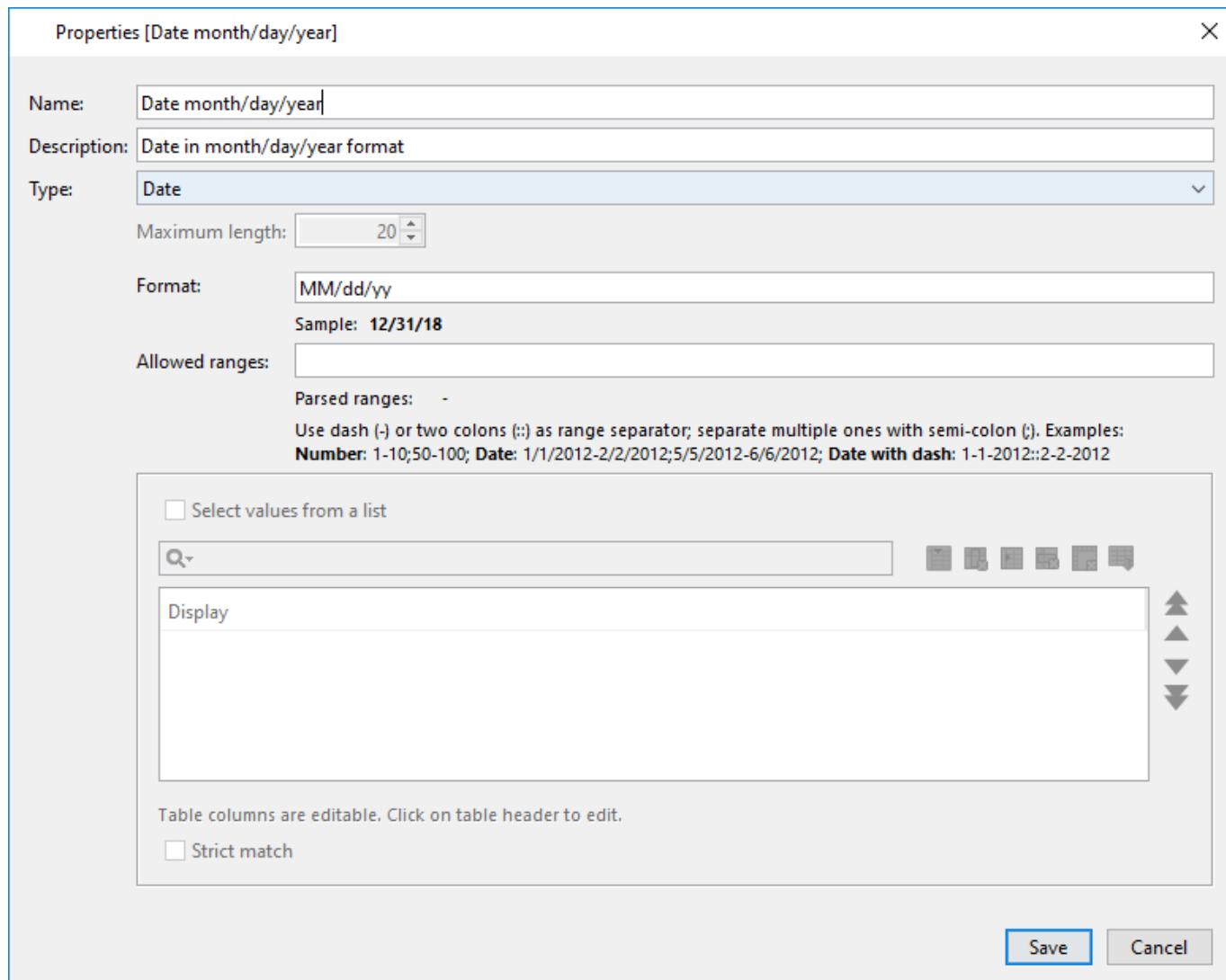


Figure 99. Field Type Properties dialog

The following properties are available:

- *Name*: the name of the field type: this is usually descriptive of the usage rather than of the underlying data types (e.g. *Address* rather than *String200*).
- *Description*: a short description of the field type
- *Type*: this is the underlying data type (domain) of the field type. Data types are matched to database data types according to the following table. Notice that each database may not support the exact data types or name them in exactly the same manner.

Name	Database type	Range
String	CHAR	<i>Maximum length</i> Unicode characters

Name	Database type	Range
Date (generic)	DATE	Holds a <i>Date</i> , not associated with a time zone
Time	TIME	Holds a <i>Time</i> , not associated with a time zone
Time-stamp	DATETIME	Holds a Date, Time or both. Represents an instant in time.
Numeric	DECIMAL	<i>Maximum length</i> digits, including <i>Decimal places</i> decimal digits (arbitrary-precision signed decimal number)
Integer	INTEGER	32-bit signed two's complement integer. It has a minimum value of -2,147,483,648 and a maximum value of 2,147,483,647 (inclusive).
Double	DOUBLE	double-precision 64-bit IEEE 754 floating point

- *Maximum length*: this field is enabled for the *String* and *Numeric* data types. For *String* data types, it represents the maximum number of allowed Unicode characters. For *Numeric* data types, it is the maximum number of digits.
- *Decimal places*: this field is enabled for the *Numeric* data type only. For example, the Oracle data type **NUMBER(10,2)** is equivalent to having *Maximum length* = 10 and *Decimal places* = 2 (Oracle treats the ANSI data type **DECIMAL(x,y)** as **NUMBER(x,y)**).
- *Allowed ranges*: this field allows you to enter one or more ranges of values to restrict the allowed values of the field type. It is enabled for all fields except for *String* and *Time*. You may enter more than one range of values separated with a semi-colon; you may use the dash character (-) or two colons (::) as the range separator. For example, if you want a field type that can take a value between **10** and **20** or between **50** and **60**, you can write here: **10-20;50-60** or **10::20;50::60**. For *Date* data types, the format of the date depends on the *Format* property right below: if you have specified it, then the format of the dates you enter here must follow the format you specified there; if you have left the *Format* property empty, then the one from the *regional date format settings* from the *Windows Control Panel* will be used. As you type in the *Allowed ranges* text box, Info Input Solution immediately parses the ranges and switches the text color to red if it cannot recognize the pattern you entered.
- *Format*: this field is used to specify the format of the date and time for the *Date* and *Time* types with the usage of a *pattern string*. Within date and time pattern strings, unquoted letters from **A** to **Z** and from **a** to **z** are interpreted as pattern letters representing the components of a date or time string. Text can be quoted using single quotes (') to avoid interpretation. "''" represents a single quote. All other characters are not interpreted; they're simply copied into the output string during formatting. The following pattern letters are defined (all other characters from **A** to **Z** and from **a** to **z** are reserved):

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Pattern letters are usually repeated, as their number determines the exact presentation:

- Text: for formatting, if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available. For parsing, both forms are accepted, independent of the number of pattern letters.##
- Number: For formatting, the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it's needed to separate two adjacent fields.##
- Year:
 - For formatting, if the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a number.##
 - For parsing, if the number of pattern letters is more than 2, the year is interpreted literally,

regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.

- ° For parsing with the abbreviated year pattern ("y" or "yy"), Info Input Solution must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the current time. ##
- Month: If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.
- RFC 822 time zone: the RFC 822 4-digit time zone format is used.##

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2001-07-04T12:08:56.235-0700

- the *Sample* label right below the *Format* text box provides a real-time date/time sample according to the date/time pattern you type to help you visualize the date and at the same time catch any errors.
- *Select values from a list*: this option is only enabled for *String* data types. If you select it, then you will need to specify the list of available values for this field type.

3.2.3.2. Using Dates

The *DATE* and *TIME* types are not associated with a time zone, whereas the *TIMESTAMP* really represents an instant in time. The *TIMESTAMP* type is a good choice if you want to represent the exact time of an event on the time-line. The *DATE* is a good choice if you want to represent a "fixed" date, like a date of birth, where this date is constant throughout the world. The *TIME* type represents information about the time of a local event, like the time of day that a shop opens or closes.

TIMESTAMP types are always displayed and parsed in the current end-user local time zone. *DATE* and *TIME* types are always displayed and parsed in a uniform manner, no matter the local time zone. Let's

take as an example two users: one is New York, where the time zone is GMT-5 and one in Brussels where the time zone is GMT+1. The following table summarizes what happens if the user in NY types a date/time in a field and the user in Brussels tries to view it (all dates are in M/D/Y format; Brussels is 6 hours ahead of New York).

Date Type	input from NY user	output for Brussels user	output for New York user
DATE	3/11/1975	3/11/1975	3/11/1975
TIME	09:00:00	09:00:00	09:00:00
TIMESTAMP	3/11/1975	3/11/1975	3/10/1975
TIMESTAMP	15:00:00	09:00:00	15:00:00
TIMESTAMP	3/11/1975 18:00:00	3/11/1975 12:00:00	3/11/1975 18:00:00
TIMESTAMP	3/11/1975 18:00:00	3/11/1975 00:00:00	3/10/1975 18:00:00

Notice that the *TIMESTAMP* type may represent a date, a time or both: what differentiates it from the *DATE* and *TIME* types is that it does represent a specific instant in time by storing the user's local time zone, the value will change when being displayed according to the viewer's time zone. On the other hand, the *DATE* and *TIME* types, will always display the same value regardless the time zone where the values were loaded or saved.

DATE, *TIME* and *TIMESTAMP* data types support ranges: when defining the type, you may add one or more ranges to limit their values. Ranges are time zone agnostic for *DATE* and *TIME* types, and time zone specific for *TIMESTAMP* types. For example, if an administrator in New York defines a *TIMESTAMP* type and adds as a valid range from 3/11/1975 00:00:00 until 3/12/1975 00:00:00, a user in Brussels will be able to input values between 3/11/1975 06:00 and 3/12/1975 06:00. Notice that the value 3/11/1975 00:00:00 will not be between the limits, if entered by a Brussels user, whereas it will be accepted if entered from a NY user.

3.2.3.3. Using lists

For *String Field Types*, it is possible to define a list of values. In this case, while indexing the corresponding *Index Field*, the user will see a drop-down list and s/he can choose to pick a value from the list. If you check the *Strict match* option, then the user will only be allowed to select a value from the list. If the *Strict match* option is not checked, then the user will be able to enter any value.

Click on the Insert (or ALT+I) button to add a new row in the table. By default, for each item in the list, there are 2 properties you can define, the *Value* and the *Description*: the *Description* is what the user sees, and the *Value* is an internal property/ID that will be exported at the end. If you don't need to have 2 different properties (e.g. a different 'display' value and a different 'commit' value), you can only fill in one of the two properties (either the *Value* or the *Description*), or you can delete any of those columns.

Properties [Color]

Name:	Color										
Description:	List of colors										
Type:	String										
Maximum length:	20										
Format:											
Sample:	-										
Allowed ranges:											
Parsed ranges: -											
Use dash (-) or two colons (::) as range separator; separate multiple ones with semi-colon (;). Examples: Number: 1-10;50-100; Date: 1/1/2012-2/2/2012;5/5/2012-6/6/2012; Date with dash: 1-1-2012::2-2-2012											
<input checked="" type="checkbox"/> <u>Select values from a list</u> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Color-ID</th> <th style="width: 70%;">Display</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Red</td> </tr> <tr> <td>2</td> <td>Green</td> </tr> <tr> <td>3</td> <td>Blue</td> </tr> <tr> <td>.</td> <td>.</td> </tr> </tbody> </table>		Color-ID	Display	1	Red	2	Green	3	Blue	.	.
Color-ID	Display										
1	Red										
2	Green										
3	Blue										
.	.										
Table columns are editable. Click on table header to edit. <input type="checkbox"/> Strict match											
<input type="button" value="Save"/> <input type="button" value="Cancel"/>											

Figure 100. Field Type with a list of possible values

In the example above, we define a *Field Type* named 'Color' and we have provided 4 possible values: red, blue, black and white. Note that for convenience, the column names of the table can be edited by double clicking on the column headers. We have named the first column 'Color-ID' and it contains an integer representing each color. In the following picture you can see how to define an *Index Field* of type 'Color'.

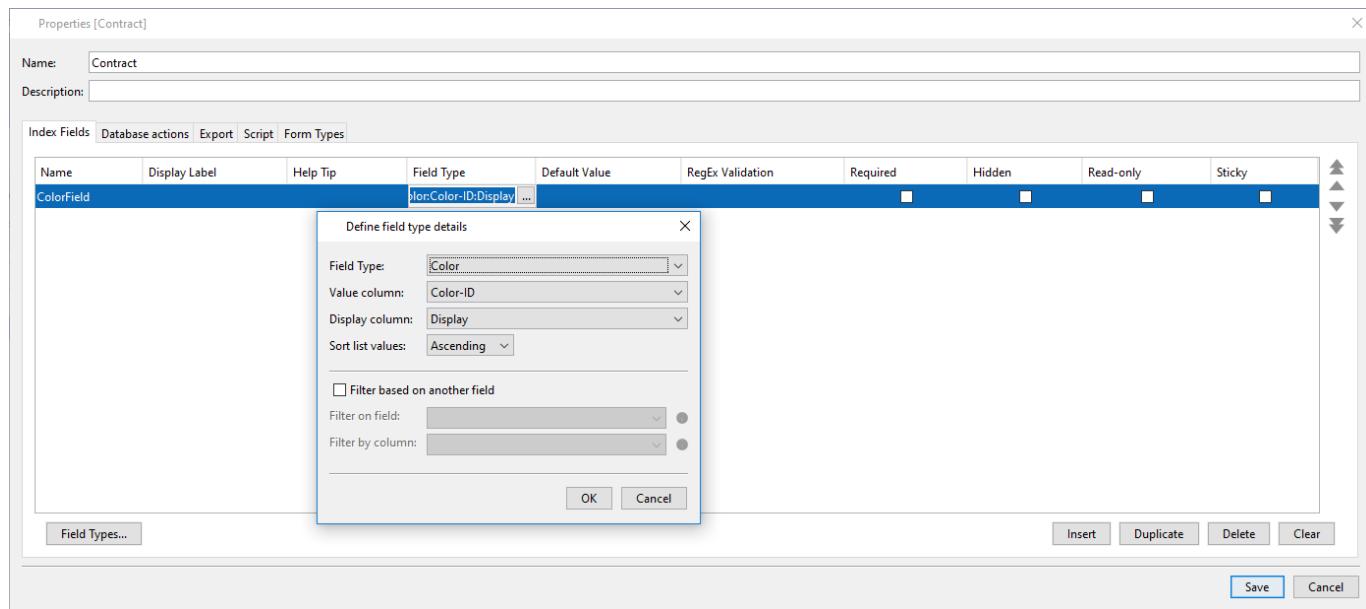


Figure 101. Defining an Index Field of a Field Type that uses a list of values

In the *Define field type details* dialog, when you select a *Field Type* that contains a list of values, the *Value column* and *Display column* combo boxes are enabled and they are filled with the column names that you have used in the *Field Type* definition. You need to select which column will be used as the *Value* of the *Index Field* and which will be the *Display* value that will be visible to the user. It is not mandatory to assign both the *Value* and the *Display* columns, but if you define only one of them, then it is assumed the the display and the commit value of the *Index Field* will be the same.

Multiple level lists

Info Input Solution allows you to have *Index Fields* with a list of possible values, which is not statically defined, but it changes dynamically based on the value of another *Index Field*. This can be achieved with the use of a *Field Type* that has a table of all possible values.

Let's consider an example of three *Index Fields* named 'State', 'City' and 'Road'. The State field will display a drop-down list of all States. Based on the selected State, the drop-down list for the City field will change dynamically to show only the cities that belong to that State. Similarly, based on the selected city, the drop-down list for the Road field will change dynamically to show the roads of that city. There is no limit to the number of levels in such a hierarchy of *Index Fields*.

Name: State-City-Road

Description: A list of states, cities, roads

Type: String

Maximum length: 50 Decimal places: 2

Allowed ranges:

Use dash (-) or two colons (::) as range separator; separate multiple ones with semi-colon (;). Examples:
Number: 1-10;50-100; **Date:** 1/1/2012-2/2/2012;5/5/2012-6/6/2012; **Date with dash:** 1-1-2012::2-2-2012

Format: (leave empty for locale specific)

Sample: -

Select values from a list

Strict match

State	State Name	City-Code	City	Road	Road-Code
AZ	Arizona	1	Phoenix	Arizona Ave.	1001
AZ	Arizona	1	Phoenix	Broadway Ave.	1002
NY	New York	9	New York	West 1st st.	1003
NY	New York	9	New York	West 14th st.	1004
NY	New York	9	New York	Broadway Ave.	1005
NY	New York	9	New York	West 32nd st.	1006
NY	New York	10	Albany	Niagara Ave.	1007
IL	Illinois	60	Chicago	Lake Shore Dr.	1008
IL	Illinois	60	Chicago	Winsconsin Ave.	1009
IL	Illinois	61	Champaign	Green st.	1010

Table columns are editable. Click on table header to edit.

Figure 102. Field Type definition for states, cities and roads

To achieve this functionality, we define a *Field Type* named 'State-City-Road' as shown in the picture above. Note that the data in the table can be entered manually or imported from a text file.

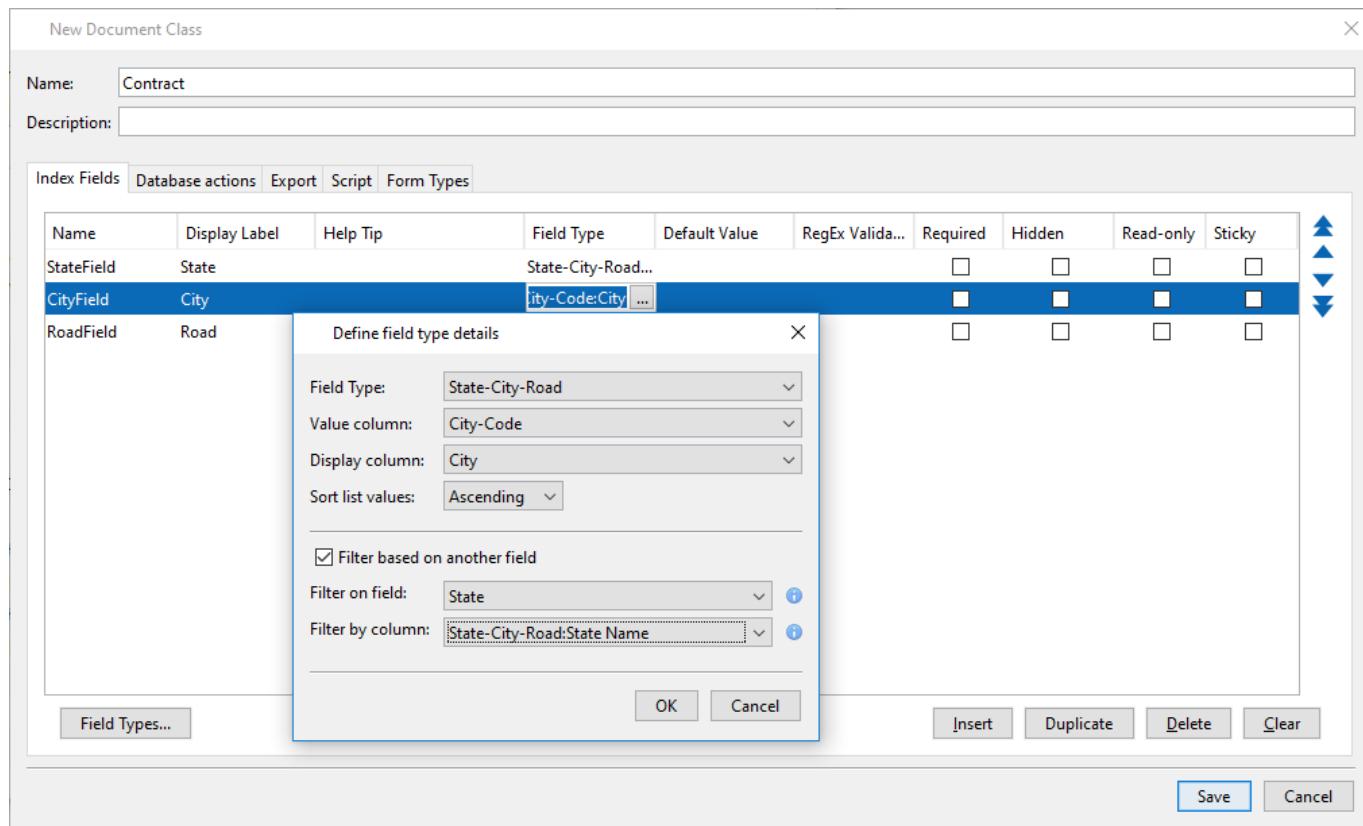


Figure 103. Error when trying to publish a Job with invalid Index Field due to Field Type column deletion

In the picture above, we show how to define the three *Index Fields*: State, City and Road. All three *Index Fields* use the same *Field Type* 'State-City-Road'. The State field uses column 'State' as the commit value and column 'State Name' as the display value. The City field uses column 'City-Code' as the commit value and column 'City' as the display value. In addition, the possible values of the City field are filtered based on the display value of the State field. The filtering is done by looking up the display value of the State field into column 'State Name' of the 'State-City-Road' Field Type. So, if the user selects New York in the State Field, then the drop-down list for City field will have New York and Albany and they will be sorted in ascending order. Similarly we define the Road field to have its values depend on the City field. Note that the *filter field* need not be a field of the same type, but it may be any other field of any type.

Note that if you define a chain of *Index Fields* where one depends on the value of another and a loop is detected, then an error will be shown when you try to save them.

Modifying Field Types with lists

It is possible to modify an existing *Field Type* and to delete one or more column from the table of possible values. However, this might invalidate *Index Fields* which are using this *Field Type* and are referencing columns in the table of possible values. If you try to save a *Document Class*, *Folder Class* or *Job* that contains an *Index Field* which was invalidated then a warning message will be shown. For example, if you delete columns Road and Road-Code in *Field Type* 'State-City-Road', then a warning message like

the following will appear if you try to save the *Index Class* that uses State-City-Road':

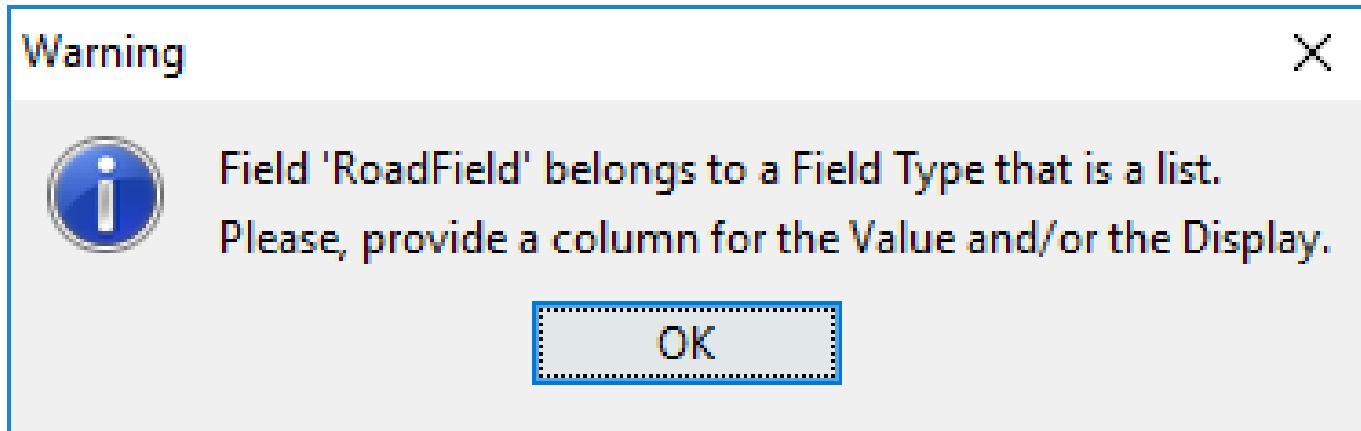


Figure 104. Error when trying to publish a Job with invalid Index Field due to Field Type column deletion.

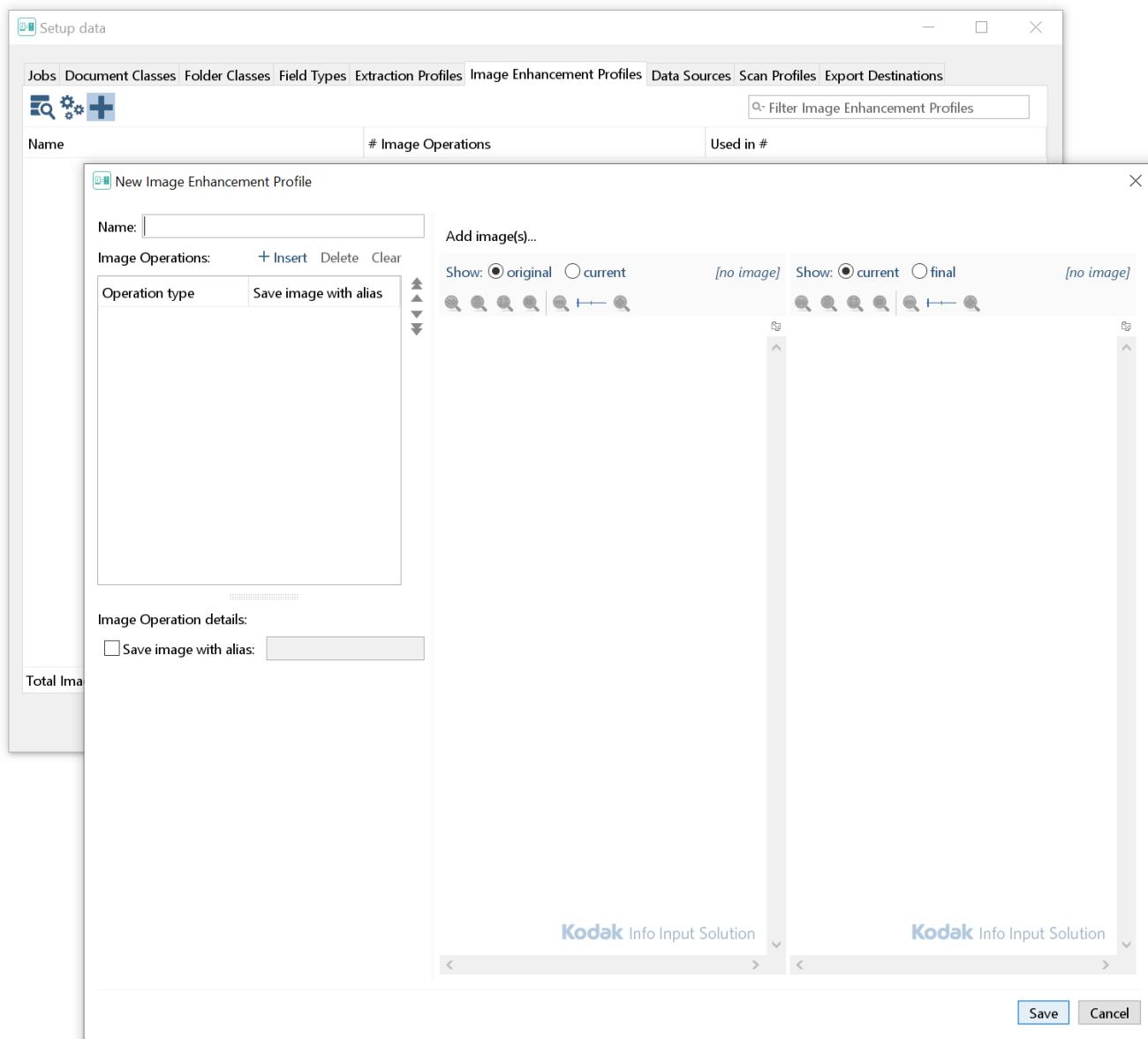
3.2.3.4. Deleting a Field Type

In order to delete a *Field Type* it must not be used by any *Index Field* at *Batch level indexing* or of any *Document* or *Folder Class*.

3.2.4. Image Enhancement profiles

The *Image Enhancements Profiles* can be used to perform any image operation on the batch images. One or more *Image Operations* can be defined in each *Enhancements Profile*, the defined *Image Operations* will be executed sequentially in the order that they are defined in the *Image Operations* table (see below). The order in which the *Image Operations* are going to be executed is from the top to bottom.

The *Image Enhancement Profiles* can be used in an *Image Enhancement Workflow* step. The *Image Enhancement Profiles* can be created from the *Job Setup* → *Image Enhancement Profiles* by pressing the *Plus* button or from within an *Image Enhancement Workflow* step by selecting the button *Image Enhancement Profiles...* See the image below,



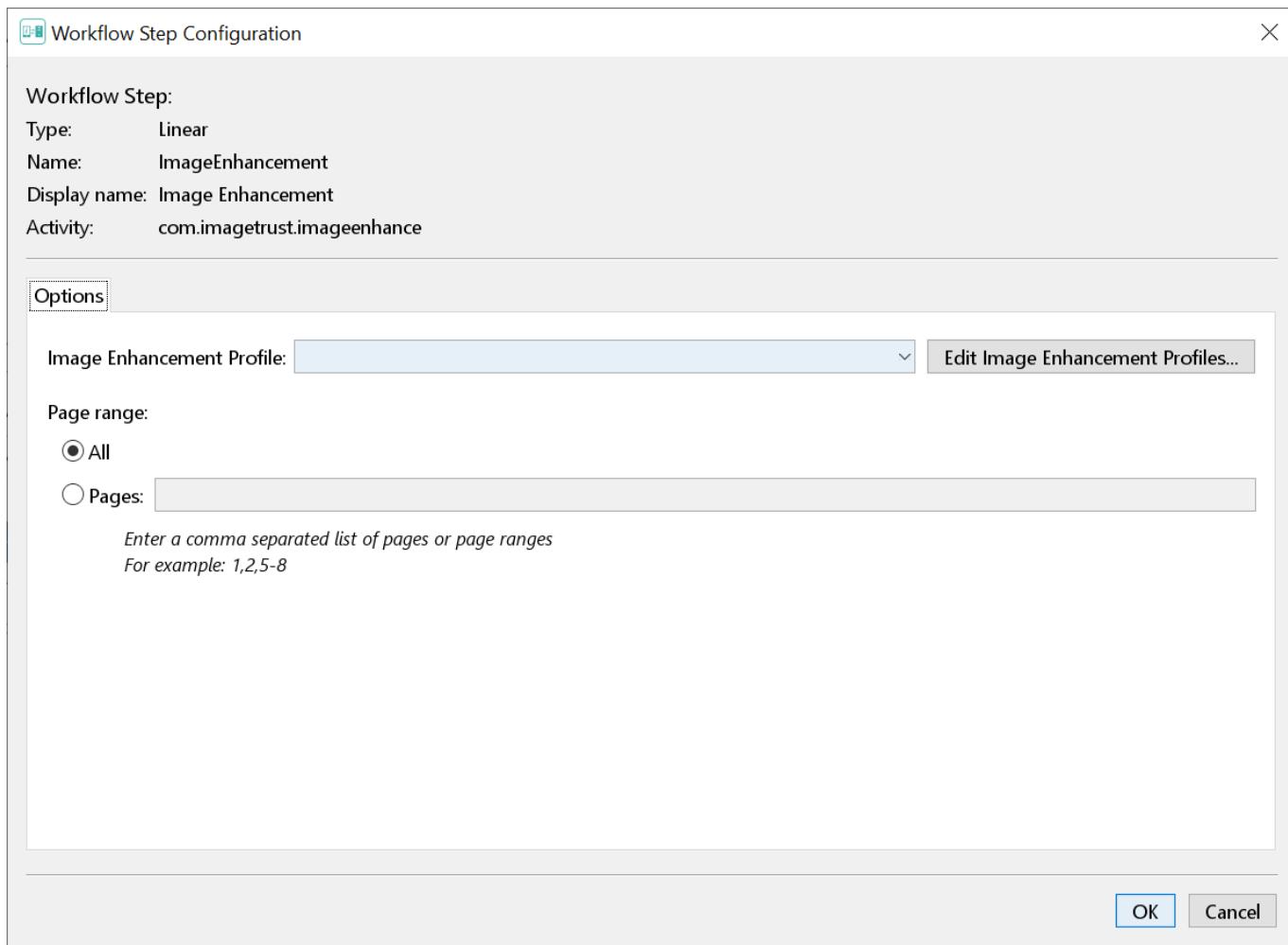


Figure 105. Image Enhancement Profiles from Job Setup and from Workflow configuration step

The following options are available in the Image Enhancements Profile configuration,

- **Name:** this will be the name of the Image Enhancements Profile.
- **Image Operations:** This is a table that shows all the Image operations that are currently defined. Select the button Insert to add a new Image operation, the Delete to delete an Image operation and Clear to remove all Image Operations. The Image Operations are described in the section [Image Operations](#).
- **Image Operation arrows up and down:** These arrows can be used to set which image operation will be applied first. The order is from top to bottom.
- **Image Operation details:** This section provide further configuration for the image operation that is highlighted. For example, for the operation Convert to Black and White the Binarization method and Binarization threshold options will be available. See the image below,

Image Operation details:

Save image with alias:

Binarization method:

Binarization threshold:



127 

Converts the image to black and white using one of 3 methods. Applies to color and grayscale images only.

- *Add Images:* This button can be used to add one or multiple images in the two viewers. See the image below,

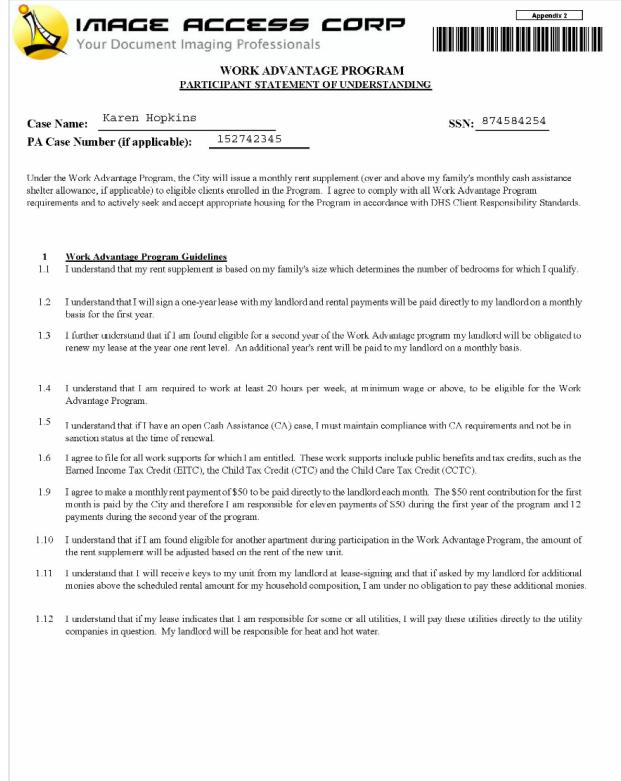
Add image(s)...

Housing Applications Sm BC_Page_01 #1 × Housing Applications Sm BC_Page_02 #1 × Housing Applications Sm BC_Page_03 #1 ×

Show: original current [original] Show: current final [original]

Convert to black and white





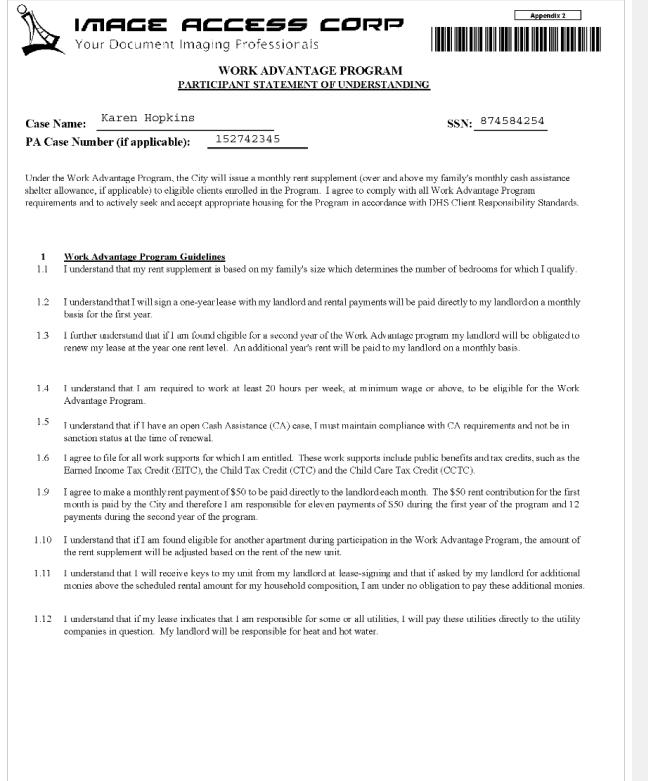


IMAGE ACCESS CORP
Your Document Imaging Professionals

WORK ADVANTAGE PROGRAM
PARTICIPANT STATEMENT OF UNDERSTANDING

Case Name: Karen Hopkins SSN: 874584254
PA Case Number (if applicable): 152742345

Under the Work Advantage Program, the City will issue a monthly rent supplement (over and above my family's monthly cash assistance shelter allowance, if applicable) to eligible clients enrolled in the Program. I agree to comply with all Work Advantage Program requirements and to actively seek and accept appropriate housing for the Program in accordance with DHS Client Responsibility Standards.

1 Work Advantage Program Guidelines

1.1 I understand that my rent supplement is based on my family's size which determines the number of bedrooms for which I qualify.

1.2 I understand that I will sign a one-year lease with my landlord and rental payments will be paid directly to my landlord on a monthly basis for the first year.

1.3 I further understand that if I am found eligible for a second year of the Work Advantage program my landlord will be obligated to renew my lease at the year one rent level. An additional year's rent will be paid to my landlord on a monthly basis.

1.4 I understand that I am required to work at least 20 hours per week, at minimum wage or above, to be eligible for the Work Advantage Program.

1.5 I understand that if I have an open Cash Assistance (CA) case, I must maintain compliance with CA requirements and not be in sanction status at the time of renewal.

1.6 I agree to file for all work supports for which I am entitled. These work supports include public benefits and tax credits, such as the Earned Income Tax Credit (EITC), the Child Tax Credit (CTC) and the Child Care Tax Credit (CCTC).

1.9 I agree to make a monthly rent payment of \$50 to be paid directly to the landlord each month. The \$50 rent contribution for the first month is paid by the City and therefore I am responsible for eleven payments of \$50 during the first year of the program and 12 payments during the second year of the program.

1.10 I understand that if I am found eligible for another apartment during participation in the Work Advantage Program, the amount of the rent supplement will be adjusted based on the rent of the new unit.

1.11 I understand that I will receive keys to my unit from my landlord at lease-signing and that if asked by my landlord for additional monies above the scheduled rental amount for my household composition, I am under no obligation to pay these additional monies.

1.12 I understand that if my lease indicates that I am responsible for some or all utilities, I will pay these utilities directly to the utility companies in question. My landlord will be responsible for heat and hot water.

1 Work Advantage Program Guidelines

1.1 I understand that my rent supplement is based on my family's size which determines the number of bedrooms for which I qualify.

1.2 I understand that I will sign a one-year lease with my landlord and rental payments will be paid directly to my landlord on a monthly basis for the first year.

1.3 I further understand that if I am found eligible for a second year of the Work Advantage program my landlord will be obligated to renew my lease at the year one rent level. An additional year's rent will be paid to my landlord on a monthly basis.

1.4 I understand that I am required to work at least 20 hours per week, at minimum wage or above, to be eligible for the Work Advantage Program.

1.5 I understand that if I have an open Cash Assistance (CA) case, I must maintain compliance with CA requirements and not be in sanction status at the time of renewal.

1.6 I agree to file for all work supports for which I am entitled. These work supports include public benefits and tax credits, such as the Earned Income Tax Credit (EITC), the Child Tax Credit (CTC) and the Child Care Tax Credit (CCTC).

1.9 I agree to make a monthly rent payment of \$50 to be paid directly to the landlord each month. The \$50 rent contribution for the first month is paid by the City and therefore I am responsible for eleven payments of \$50 during the first year of the program and 12 payments during the second year of the program.

1.10 I understand that if I am found eligible for another apartment during participation in the Work Advantage Program, the amount of the rent supplement will be adjusted based on the rent of the new unit.

1.11 I understand that I will receive keys to my unit from my landlord at lease-signing and that if asked by my landlord for additional monies above the scheduled rental amount for my household composition, I am under no obligation to pay these additional monies.

1.12 I understand that if my lease indicates that I am responsible for some or all utilities, I will pay these utilities directly to the utility companies in question. My landlord will be responsible for heat and hot water.

IMAGE TRUST

The image viewers will be updated according to the image operation that is selected from the Image Operations table.

- The left viewer displays the Image before the selected *Image Operation* has been applied. The viewer on the right display the image after applying the selected *Image Operation* has been applied. From the left viewer the option *original* will display the original Image that was scanned. The option *current* will display the image after all the *Image Operations* before the selected one have applied.
- The right viewer will display the image after the selected *Image Operation* has been applied. The option *current* will display the changes that are applied on this step only. The option *final* will display the changes after all the *Image Operations* before the selected one have applied.

3.2.4.1. Image Operations

This section describes the *Image Operations*. The operations are grouped into 7 categories according to

functionality/type:

Image Type Conversion

Black and white

Converts a grayscale or color image to black & white (binary) format: saves disk space and enables a few operations that operate only on binary images (such as despeckling or any morphological operations); if this operation is combined with other operations, it is recommended to be placed at the end of the list, unless you want to use one of those operations that require binary images.

Convert to grayscale

Converts a color image to grayscale. Grayscale or binary images are not affected.

Color/Shade Operations

The operations in this group pertain to color or shade corrections. As such, they apply only to color or grayscale images, with the exception of the "Invert Colors (Negative)" operation.

Adjust brightness/contrast

Adjusts the brightness and contrast (independently of each other) of a color or grayscale image.

Gamma correction

Simultaneously adjusts the brightness and contrast of a color or grayscale image using a non-linear (power) function by means of a single (gamma) parameter. Gamma can range from 0.1 to 10.0; a value of 1 does not affect the image colors/shades. Value under 1 make the image darker and increase the contrast; values above 1 make it brighter and reduce the contrast. In the sample images below, you can see the original image and then the same image with different values for gamma. Contrast stretching (explained below) is automatically performed on the image before the actual gamma correction.



Figure 106. Image Operations: Gamma Correction

S correction

This is similar to a "double gamma" correction. You specify the gamma factor and a brightness threshold: pixels above that threshold are subjected to a gamma correction with the specified gamma. Pixels under the threshold are subjected to a gamma correction with the inverse of the gamma factor. This operation is useful if you want to clean up the "whites" and emphasize the "dark" areas. Here is an example of a less-than-perfect scan and how it was improved by s correction. As with Gamma correction, contrast stretching (explained below) is automatically performed on the image before the s correction.##

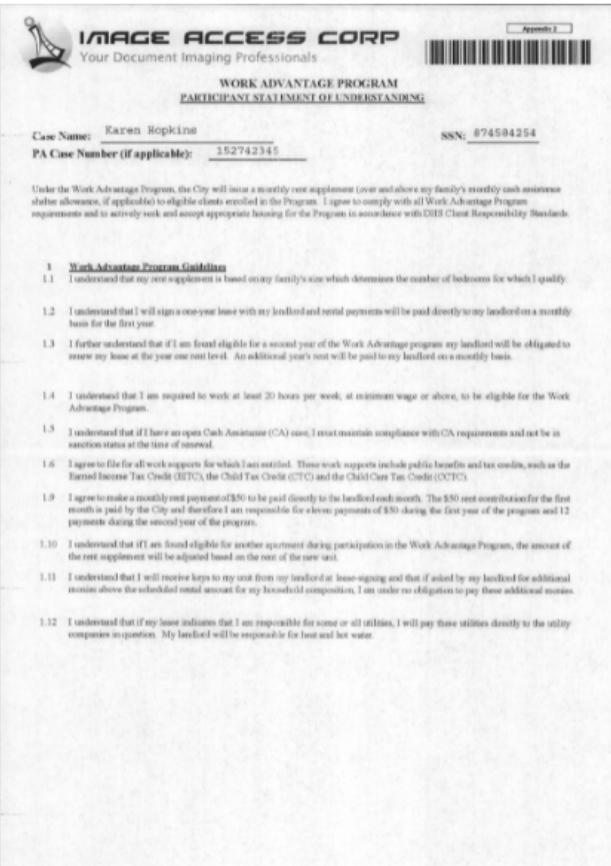
Input Image	Gamma = 10
	

Figure 107. Image Operations: S corrections example

Contrast stretch

Changes the image colors/shades as to maximize the contrast between the darkest and brightest areas. Applies to color and grayscale images only. Special care is taken so that the color balance is not affected in color images. Here are two "before/after" samples.

Input Image 1	Stretched Image 1	Input Image 2	Stretched Image 2
			

Figure 108. Image Operations: Contrast stretch

Component stretch (Auto-tone)

This operation is similar to contrast stretch, but it is applied separately to each of the 3 color channels (RGB). If the image is grayscale, then it's exactly the same as contrast stretching. The fact that the operation applies to each color component separately from the others, implies that the color balance may change; this might be the desired effect depending on the input image. Here is a "before/after" sample; observe the color shift and compare it to the above shown contrast stretching of the same image.

Input Image	Component Stretched Image
	

Figure 109. Image Operations: Component stretch

Equalize Histogram

Improves image contrast by effectively spreading out the most frequent intensity values, ie., stretching out the intensity range of the image. This allows for areas of lower local contrast to gain a higher contrast. Applies to color and grayscale images only. Here is a "before/after" sample.



Figure 110. Image Operations: Equalize Histogram

Invert Colors (Negative)

Inverts the image colors, in effect producing a negative image. It's useful in case of scanning a negative image in order to bring it to normal.



Figure 111. Image Operations: Invert Colors

Color dropout

Removes selected colors from the image: useful for cleaning up color-coded forms to facilitate subsequent extraction of the data. The color to dropout can be either specified explicitly (as Red, Orange/Brown, Yellow, Green, Cyan, Blue, Violet, Magenta) or it can be automatically selected (assuming the real form content contains only black or "near-black" or "near-gray" text). The operation applies only to color images: binary or grayscale images remain unaffected. There is a "dropout level" parameter too: it controls the aggressiveness of the dropout: it can range from -5 (least aggressive) to +5 (most aggressive) with 0 being the default. Here is a "before/after" sample.

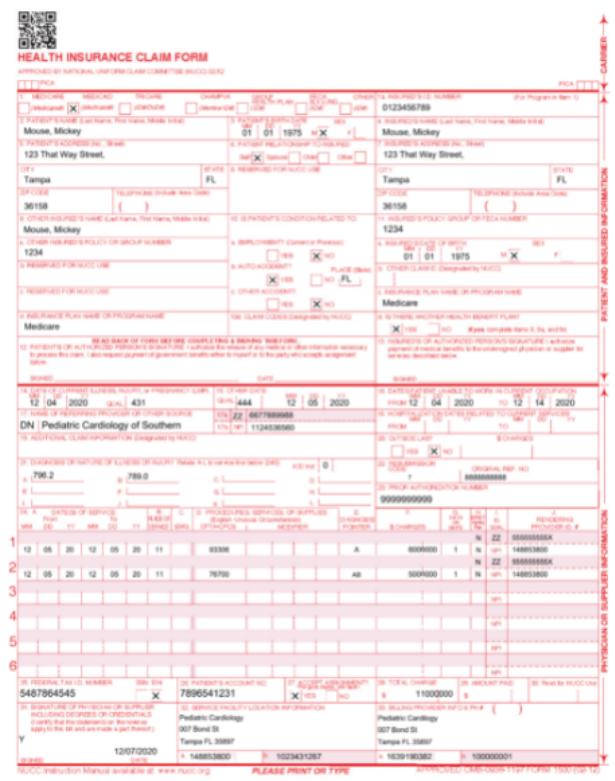
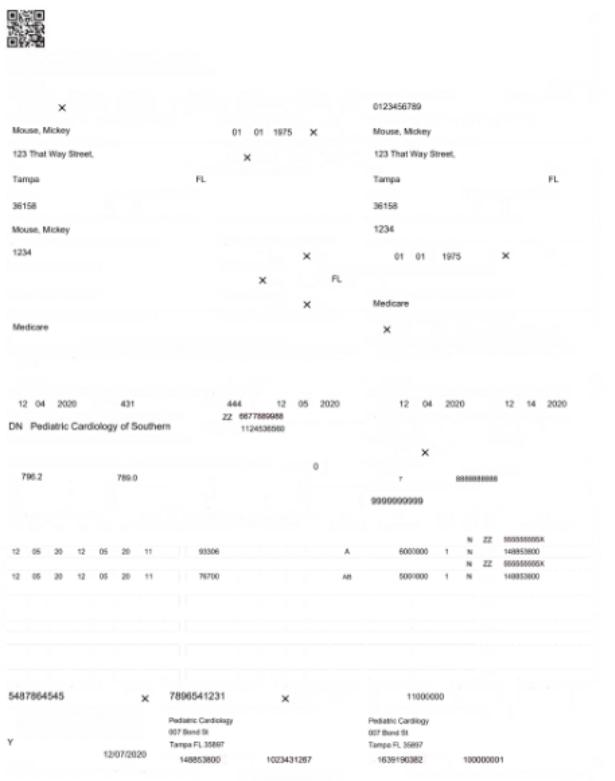
<p>Input Image</p> 	<p>Color dropped Image</p> 
--	---

Figure 112. Image Operations: Color dropout

Page Rotation

Deskew

Detects and auto-corrects skews in the scanned page ($\pm 5^\circ$). Here is a "before/after" sample.

Input Image	Deskewed Image

Figure 113. Image Operations: Deskew

Auto-rotate

Deskews the page, then detects the page rotation (normal orientation, rotated left 90°, upside down, rotated right 90°) and finally auto-rotates it so it's normally oriented. The detection is text-based: it assumes languages using either the plain or extended Latin alphabet. It does not work properly for other languages like Greek, Russian, Chinese, etc. The detection may fail for other reasons too (for example, the text in the page is too short or it's all capitals): in that case, the page is not modified. Here is a "before/after" sample:

Binary Morphology

Morphological image processing is a collection of non-linear operations related to the shape (morphology) of features in an image. The operations in this group apply only to black & white (binary) images: they do not affect images of other types. They are useful in correcting blemishes of binary images. Morphological techniques probe an image with a small shape or template called a structuring element (or kernel); the user can select the kernel size from among 3, 5, 7 and 9 pixels. ## *Erode (white thinning)*: Erodes away the boundaries of white areas. Thus, white areas shrink in size and (black) holes within those areas become larger. Here is a "before/after" sample: note the "black holes" in the eroded image.

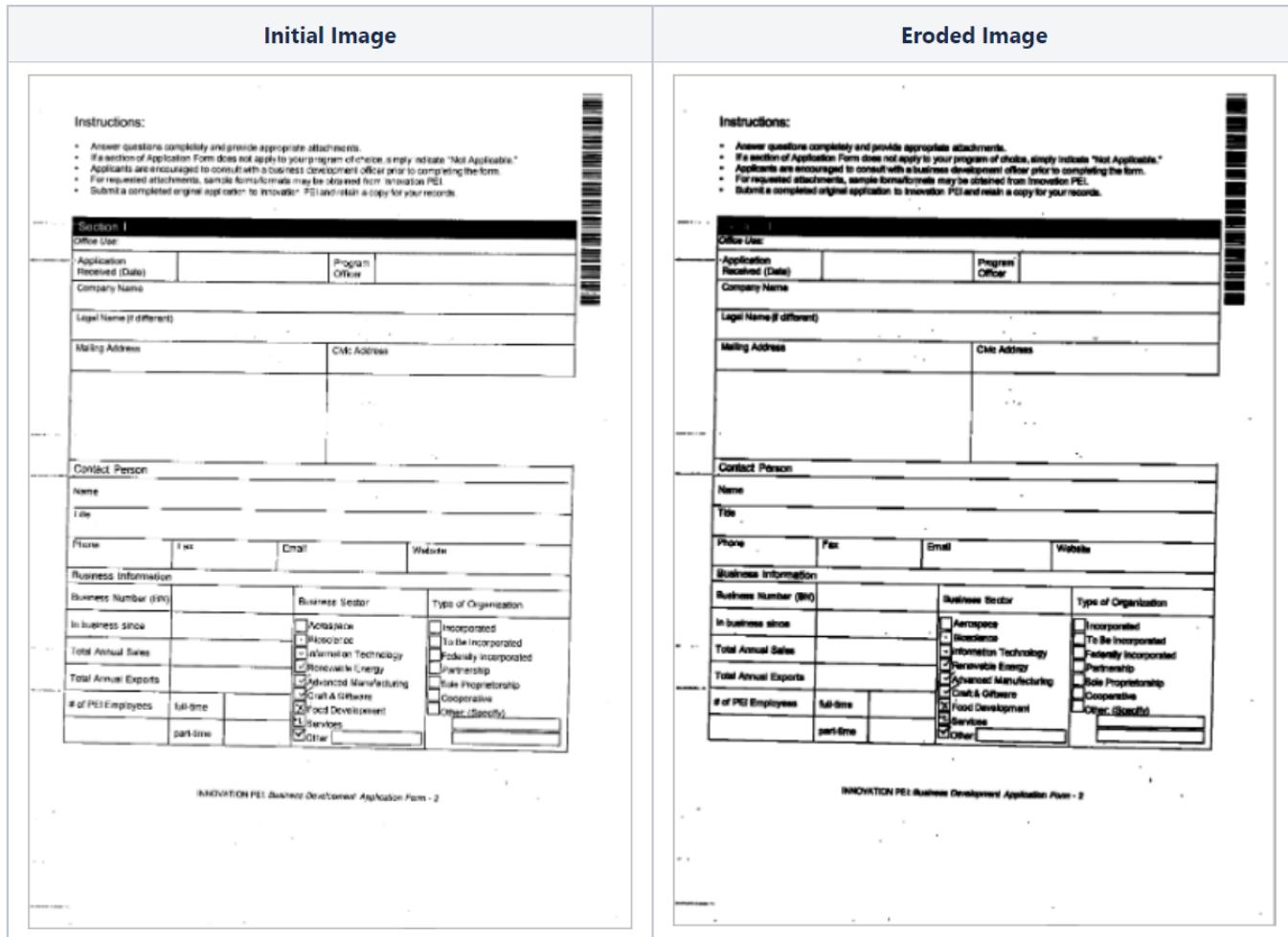


Figure 114. Image Operations: Erode

Dilate (white expansion)

Gradually enlarges the boundaries of white areas. Thus, white areas grow in size, and (black) holes within those areas become smaller. Here is a "before/after" sample: note how the black areas have thinned in the dilated image.

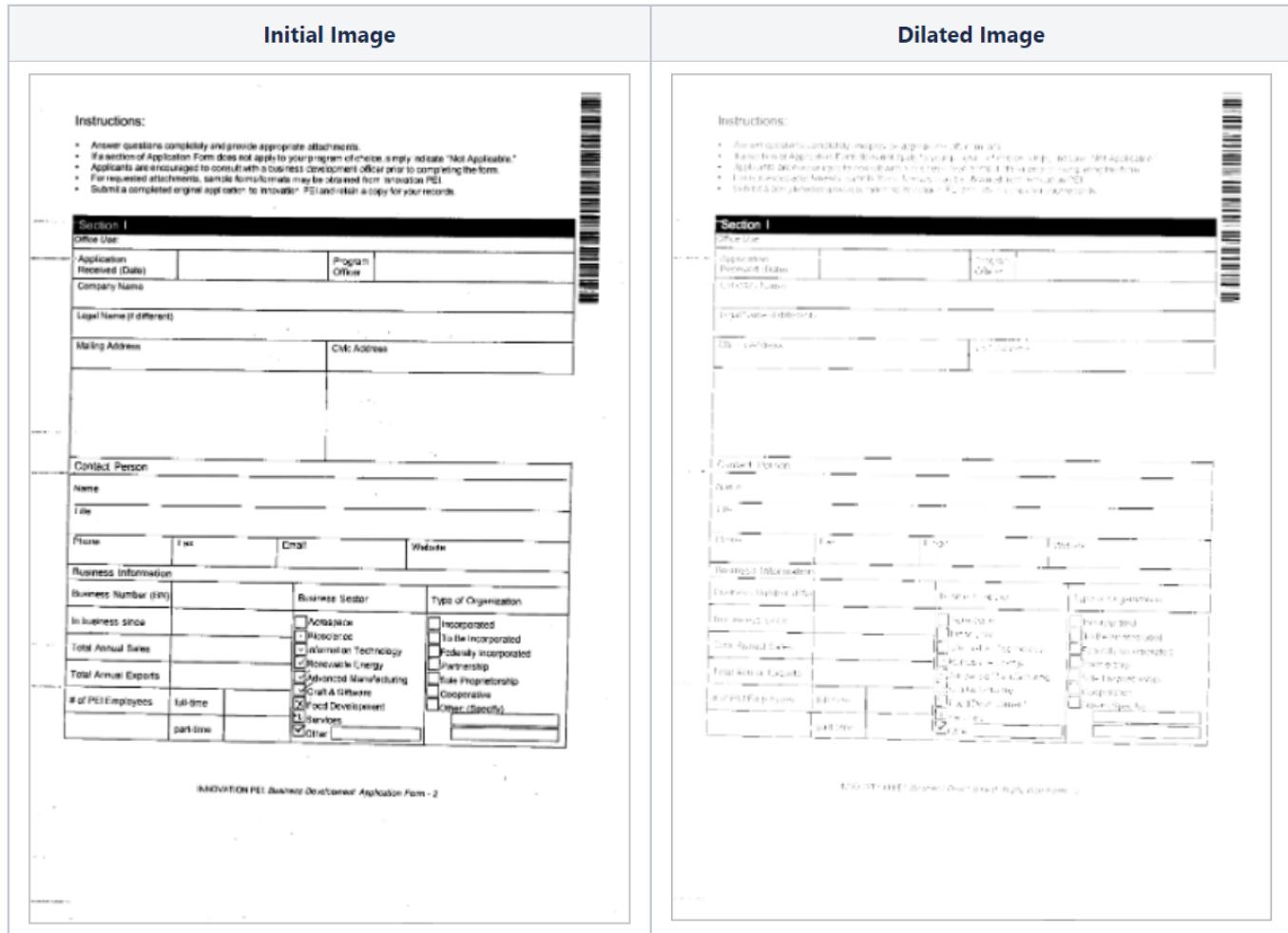


Figure 115. Image Operations: Dilate

Open (white spot removal)

Performs an erosion (reduction of white areas) followed by a dilation (enlargement of white areas). Opening removes any narrow white connections and lines between two black regions. Morphological opening is useful for removing small objects from an image while preserving the shape and size of larger objects in the image. It can also help in removing "salt noise" (small white marks).

Initial Image		Opened Image																																																																																																	
<p>Instructions:</p> <ul style="list-style-type: none"> Answer questions completely and provide appropriate attachments. If an Application Form does not apply to your program of choice, simply indicate "Not Applicable." Applicants are encouraged to consult with a program officer prior to completing the form. For requested attachments, sample forms/templates may be obtained from Innovation PEI. Submit a completed online application to Innovation PEI and retain a copy for your records. <p>Section 1 Office Use:</p> <table border="1"> <tr> <td>Application Received (Date)</td> <td>Program Officer</td> </tr> <tr> <td colspan="2">Company Name</td> </tr> <tr> <td colspan="2">Legal Name (if different)</td> </tr> <tr> <td>Mailing Address</td> <td>C/M: Address</td> </tr> <tr> <td colspan="2">Contact Person</td> </tr> <tr> <td colspan="2">Name</td> </tr> <tr> <td colspan="2">Title</td> </tr> <tr> <td>Phone</td> <td>Fax</td> </tr> <tr> <td> </td> <td>Email</td> </tr> <tr> <td colspan="2">Website</td> </tr> <tr> <td colspan="4">Business Information</td> </tr> <tr> <td>Business Number (INN)</td> <td>Business Sector</td> <td colspan="2">Type of Organization</td> </tr> <tr> <td>In business since</td> <td> <input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other </td> <td colspan="2"> <input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify) </td> </tr> <tr> <td>Total Annual Sales</td> <td></td> <td colspan="2"></td> </tr> <tr> <td>Total Annual Exports</td> <td></td> <td colspan="2"></td> </tr> <tr> <td># of PEI Employees</td> <td>Full-time</td> <td></td> <td>Part-time</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p>INNOVATION PEI Business Development Application Form - 2</p>		Application Received (Date)	Program Officer	Company Name		Legal Name (if different)		Mailing Address	C/M: Address	Contact Person		Name		Title		Phone	Fax		Email	Website		Business Information				Business Number (INN)	Business Sector	Type of Organization		In business since	<input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)		Total Annual Sales				Total Annual Exports				# of PEI Employees	Full-time		Part-time					<p>Instructions:</p> <ul style="list-style-type: none"> Answer questions completely and provide appropriate attachments. If an Application Form does not apply to your program of choice, simply indicate "Not Applicable." Applicants are encouraged to consult with a program officer prior to completing the form. For requested attachments, sample forms/templates may be obtained from Innovation PEI. Submit a completed online application to Innovation PEI and retain a copy for your records. <p>Section 1 Office Use:</p> <table border="1"> <tr> <td>Application Received (Date)</td> <td>Program Officer</td> </tr> <tr> <td colspan="2">Company Name</td> </tr> <tr> <td colspan="2">Legal Name (if different)</td> </tr> <tr> <td>Mailing Address</td> <td>C/M: Address</td> </tr> <tr> <td colspan="2">Contact Person</td> </tr> <tr> <td colspan="2">Name</td> </tr> <tr> <td colspan="2">Title</td> </tr> <tr> <td>Phone</td> <td>Fax</td> </tr> <tr> <td> </td> <td>Email</td> </tr> <tr> <td colspan="2">Website</td> </tr> <tr> <td colspan="4">Business Information</td> </tr> <tr> <td>Business Number (INN)</td> <td>Business Sector</td> <td colspan="2">Type of Organization</td> </tr> <tr> <td>In business since</td> <td> <input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other </td> <td colspan="2"> <input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify) </td> </tr> <tr> <td>Total Annual Sales</td> <td></td> <td colspan="2"></td> </tr> <tr> <td>Total Annual Exports</td> <td></td> <td colspan="2"></td> </tr> <tr> <td># of PEI Employees</td> <td>Full-time</td> <td></td> <td>Part-time</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p>INNOVATION PEI Business Development Application Form - 2</p>		Application Received (Date)	Program Officer	Company Name		Legal Name (if different)		Mailing Address	C/M: Address	Contact Person		Name		Title		Phone	Fax		Email	Website		Business Information				Business Number (INN)	Business Sector	Type of Organization		In business since	<input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)		Total Annual Sales				Total Annual Exports				# of PEI Employees	Full-time		Part-time				
Application Received (Date)	Program Officer																																																																																																		
Company Name																																																																																																			
Legal Name (if different)																																																																																																			
Mailing Address	C/M: Address																																																																																																		
Contact Person																																																																																																			
Name																																																																																																			
Title																																																																																																			
Phone	Fax																																																																																																		
	Email																																																																																																		
Website																																																																																																			
Business Information																																																																																																			
Business Number (INN)	Business Sector	Type of Organization																																																																																																	
In business since	<input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)																																																																																																	
Total Annual Sales																																																																																																			
Total Annual Exports																																																																																																			
# of PEI Employees	Full-time		Part-time																																																																																																
Application Received (Date)	Program Officer																																																																																																		
Company Name																																																																																																			
Legal Name (if different)																																																																																																			
Mailing Address	C/M: Address																																																																																																		
Contact Person																																																																																																			
Name																																																																																																			
Title																																																																																																			
Phone	Fax																																																																																																		
	Email																																																																																																		
Website																																																																																																			
Business Information																																																																																																			
Business Number (INN)	Business Sector	Type of Organization																																																																																																	
In business since	<input type="checkbox"/> Aerospace <input type="checkbox"/> Aerospace <input type="checkbox"/> Information Technology <input type="checkbox"/> Information Technology <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Specialty <input checked="" type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be Incorporated <input type="checkbox"/> Federally Incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)																																																																																																	
Total Annual Sales																																																																																																			
Total Annual Exports																																																																																																			
# of PEI Employees	Full-time		Part-time																																																																																																

Figure 116. Image Operations: Open (white spot removal)

Close (black spot removal)

Performs a dilation (enlargement of white areas) followed by an erosion (reduction of white areas). Closing fills up any narrow black regions or holes in the image. Morphological closing is useful for filling small black holes from an image while preserving the shape and size of white objects in the image. It can also help in removing "pepper noise" (small black marks).

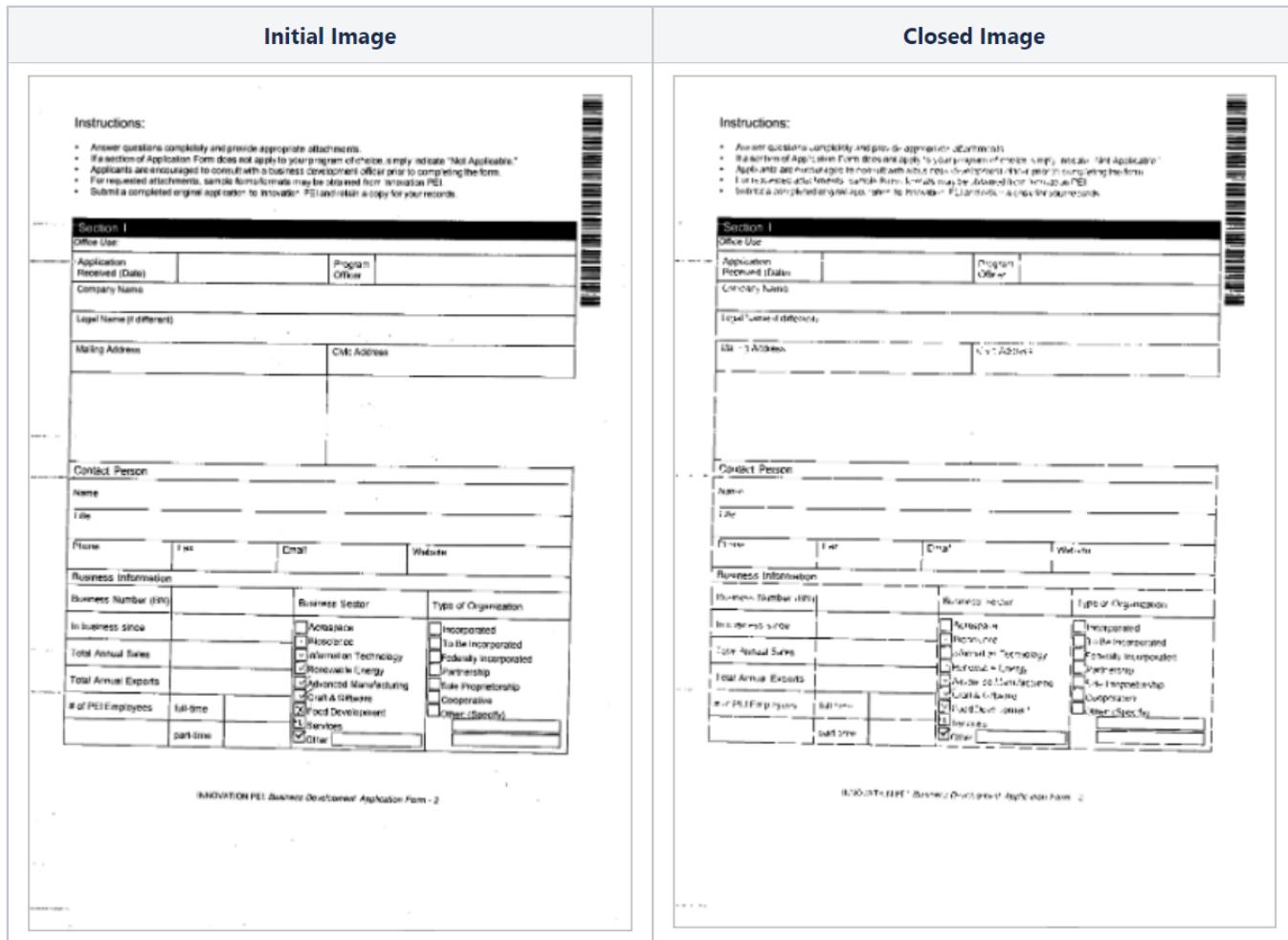
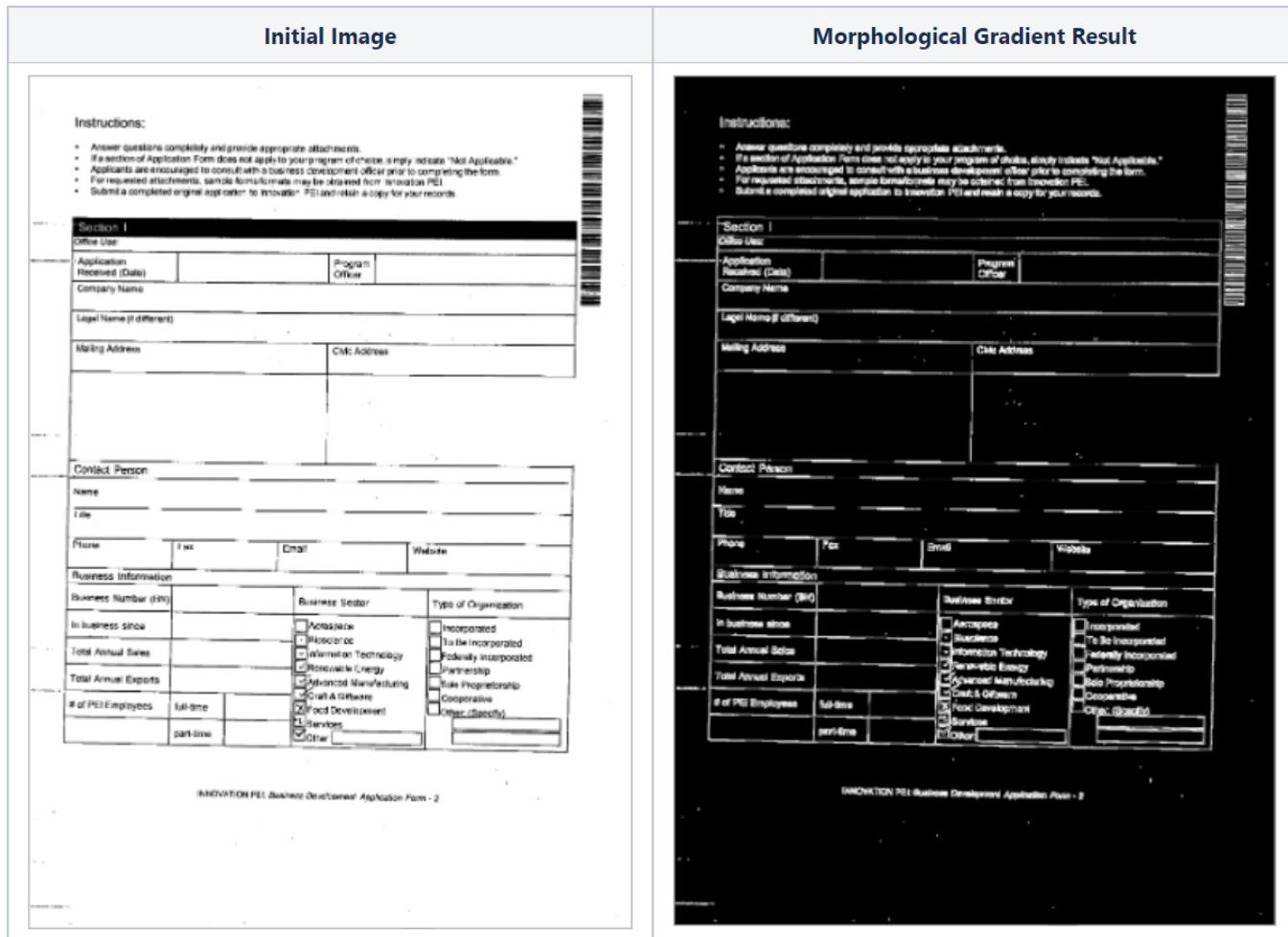


Figure 117. Image Operations: Close (black spot removal)

Morphological gradient (edge detection)

Takes the difference between the dilation and the erosion of the image. Highlights areas of high contrast. Useful for subsequent edge detection and segmentation.



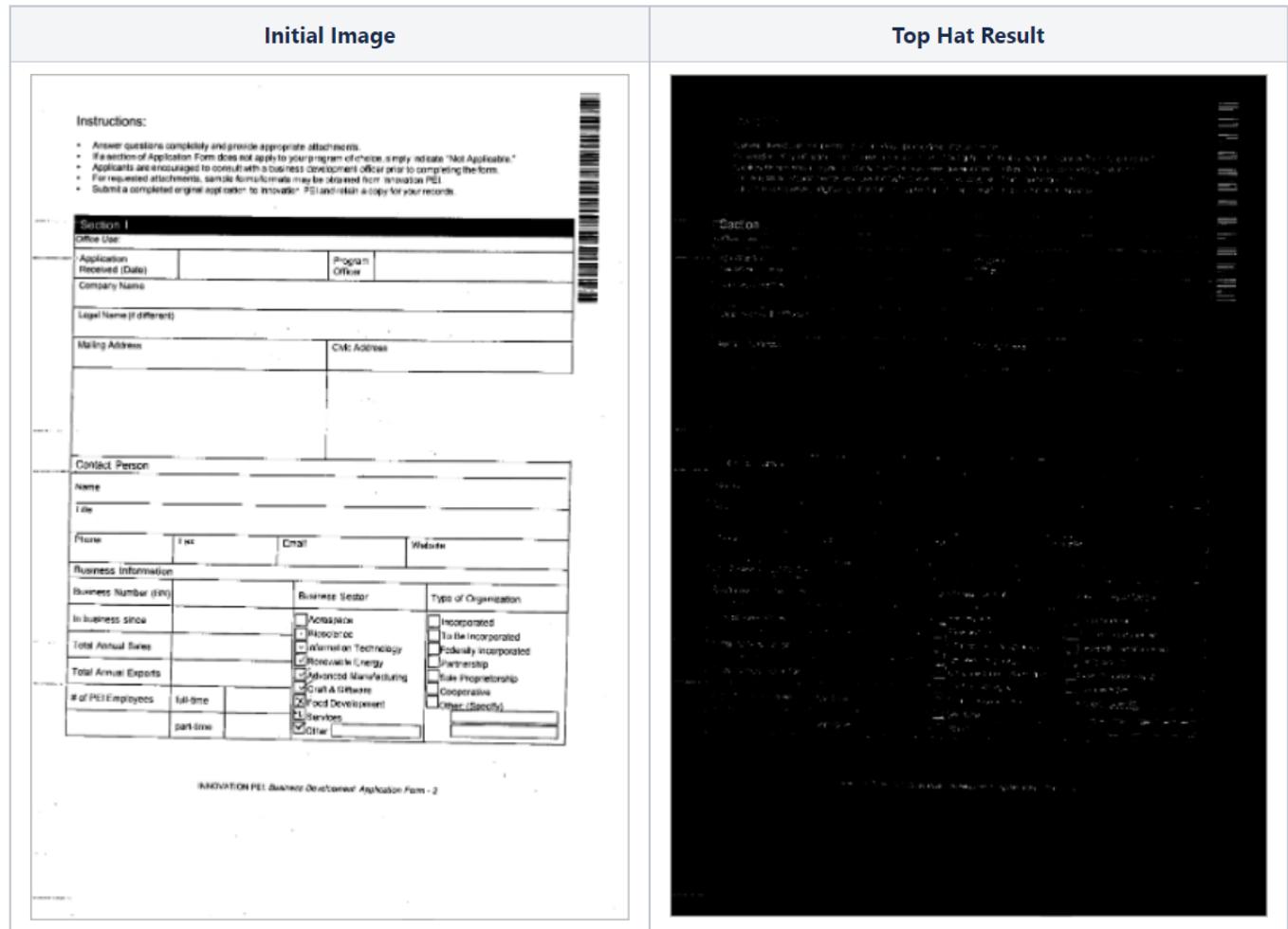


Figure 119. Image Operations: Top hat

Black hat (highlight dark spots)

The black hat transform closes an image, then subtracts the original image from the closed image. This highlights the narrow black regions in the image. Enhances dark regions on bright backgrounds.

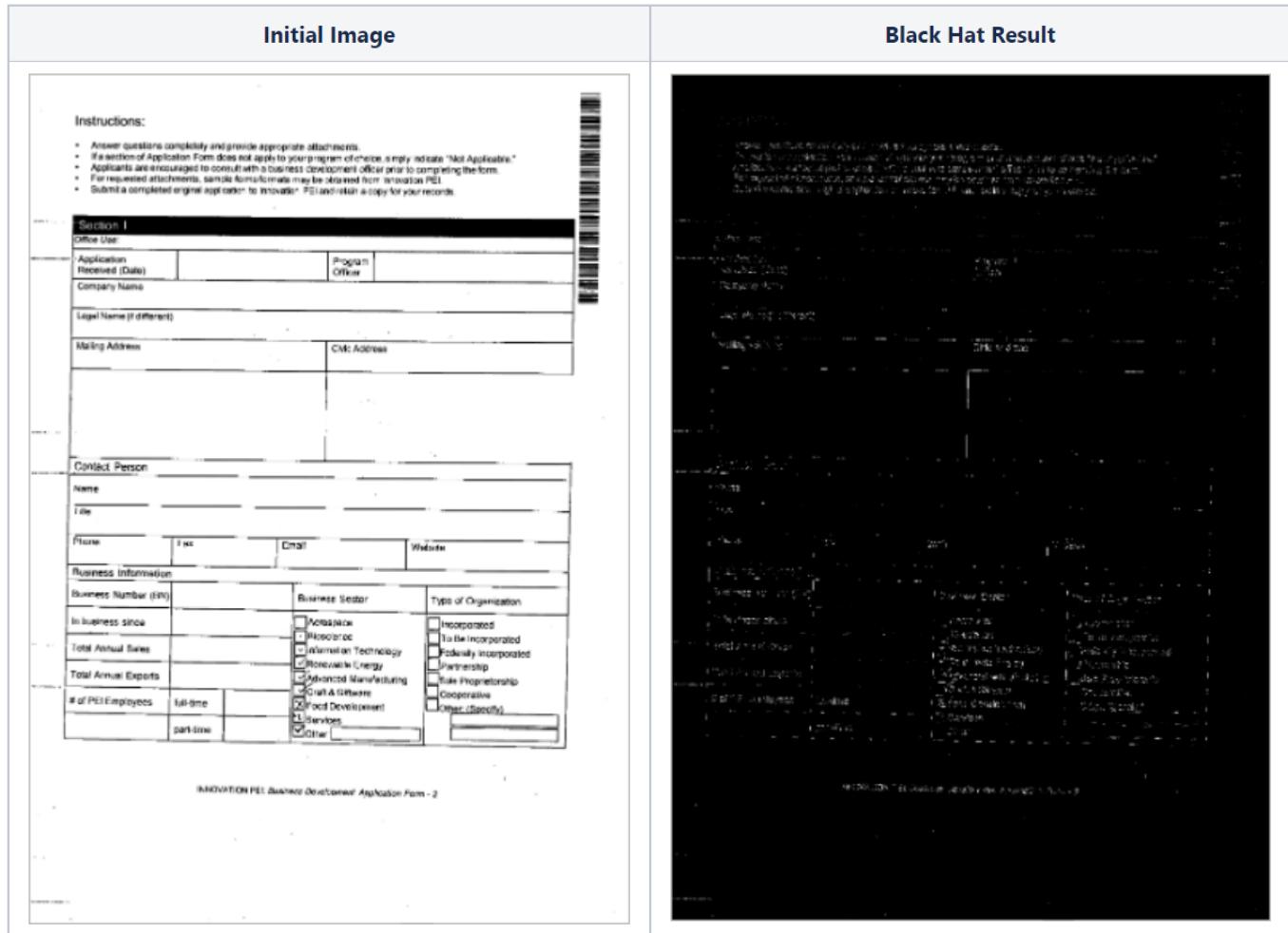


Figure 120. Image Operations: Black hat

Miscellaneous Operations

Crop

Crops the input image to the specified rectangle. The rectangle can be specified either numerically (as percentages of the page width/height) or interactively by direct selection of the desired area. Here is an image of the interactive selection process on the left, and the preview of the cropped area on the right.

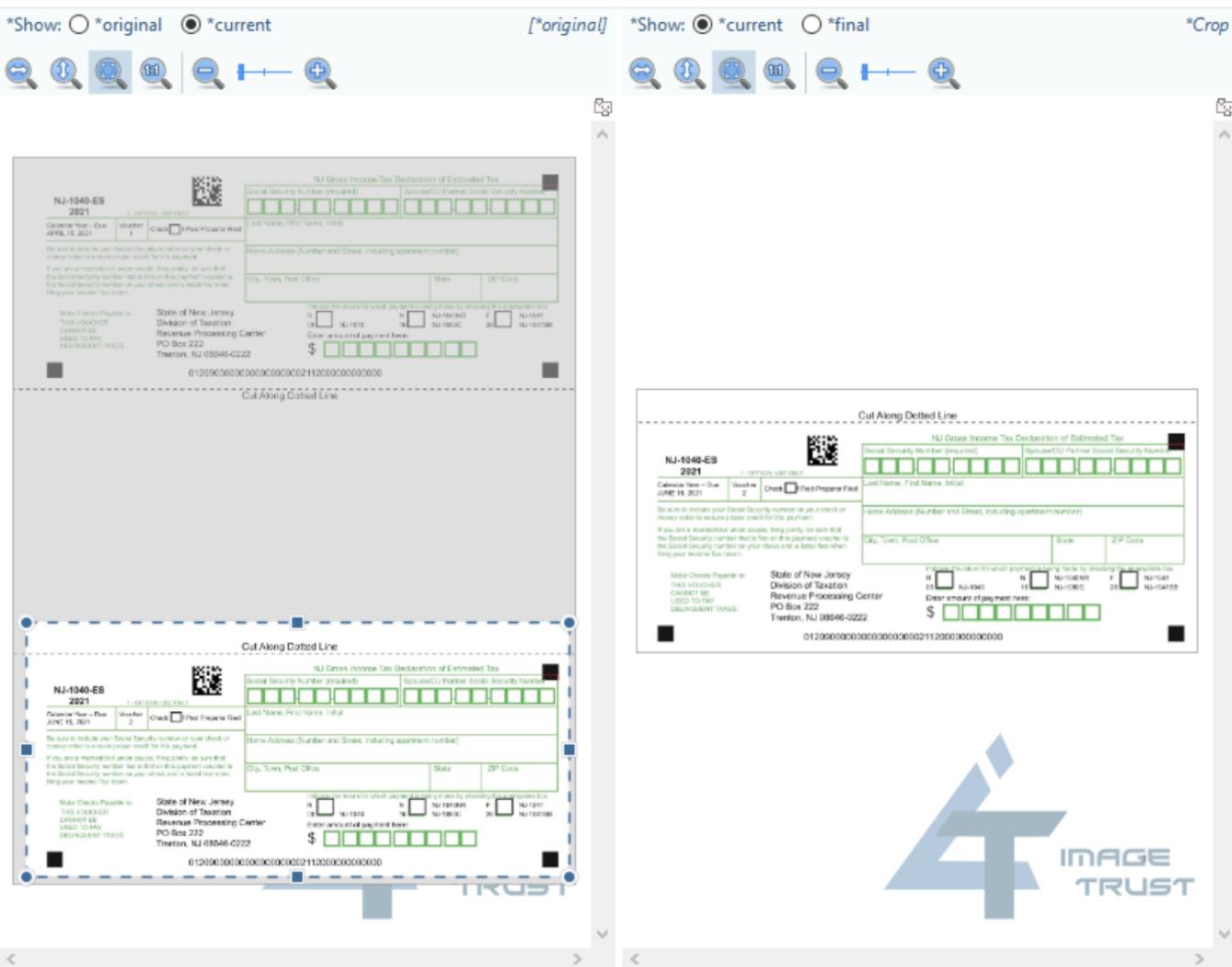


Figure 121. Image Operations: Crop

Resize:

Enlarges or shrinks the image (or alternatively, increases or decreases its resolution) by a scale factor. The scale factor can be directly entered, or it can be auto-derived if you specify the desired resolution, or the desired pixel size.

Remove black border

Detects and removes black borders from the image. If the image is in color, it is automatically converted to grayscale prior to the border removal.

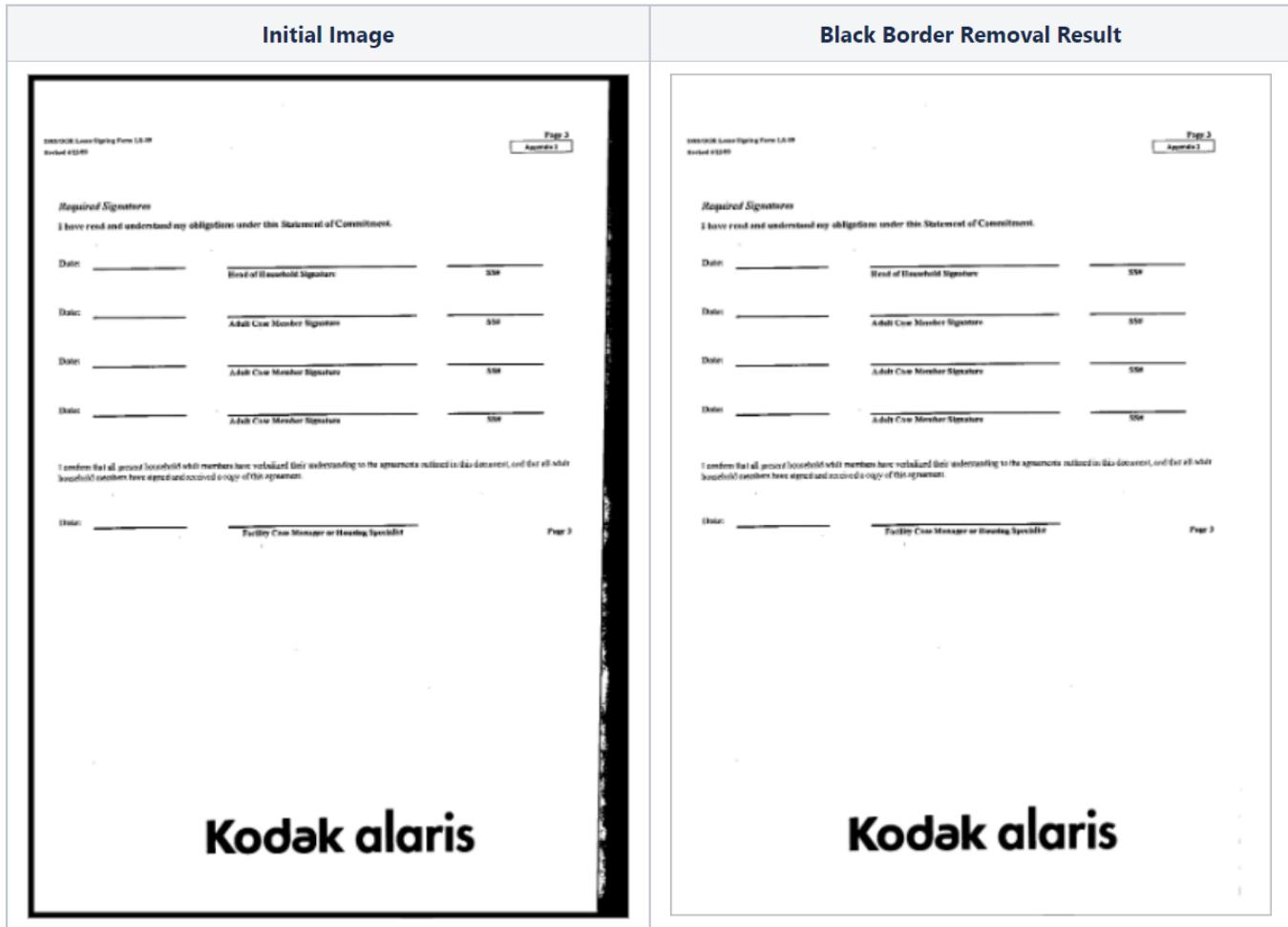


Figure 122. Image Operations: Remove black border

Despeckle (remove noise)

Removes small black marks (noise) from black & white images; it does not affect grayscale or color images.

Initial Image		Despeckled Image																																																																																													
<p>Instructions:</p> <ul style="list-style-type: none"> Answer questions completely and provide appropriate attachments. If a question does not apply, please leave a reply, indicate "Not Applicable." Applicants are encouraged to consult with their development officer prior to completing the form. For requested attachments, sample formats may be obtained from Innovation PEI. Submit a completed original application to Innovation PEI and retain a copy for your records. <p>Section I</p> <p>Office Use:</p> <table border="1"> <tr> <td>Application Received (Date)</td> <td>Program Officer</td> </tr> <tr> <td colspan="2">Company Name</td> </tr> <tr> <td colspan="2">Legal Name (if different)</td> </tr> <tr> <td>Mailing Address</td> <td>City/Address</td> </tr> <tr> <td colspan="2">Contact Person</td> </tr> <tr> <td>Name</td> <td></td> </tr> <tr> <td>Title</td> <td></td> </tr> <tr> <td>Phone</td> <td>Fax</td> <td>Email</td> <td>Website</td> </tr> <tr> <td colspan="4">Business Information</td> </tr> <tr> <td>Business Number (INN)</td> <td>Business Sector</td> <td>Type of Organization</td> <td></td> </tr> <tr> <td>In business since</td> <td colspan="3"> <input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other </td> </tr> <tr> <td>Total Annual Sales</td> <td colspan="3"></td> </tr> <tr> <td>Total Annual Exports</td> <td colspan="3"></td> </tr> <tr> <td># of PEI Employees</td> <td>Full-time</td> <td>Part-time</td> <td> <input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify) </td> </tr> <tr> <td colspan="4">INNOVATION PEI Business Development Application Form - 2</td> </tr> </table>		Application Received (Date)	Program Officer	Company Name		Legal Name (if different)		Mailing Address	City/Address	Contact Person		Name		Title		Phone	Fax	Email	Website	Business Information				Business Number (INN)	Business Sector	Type of Organization		In business since	<input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other			Total Annual Sales				Total Annual Exports				# of PEI Employees	Full-time	Part-time	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)	INNOVATION PEI Business Development Application Form - 2				<p>Instructions:</p> <ul style="list-style-type: none"> Answer questions completely and provide appropriate attachments. If a question does not apply, please leave a reply, indicate "Not Applicable." Applicants are encouraged to consult with their development officer prior to completing the form. For requested attachments, sample formats may be obtained from Innovation PEI. Submit a completed original application to Innovation PEI and retain a copy for your records. <p>Section I</p> <p>Office Use:</p> <table border="1"> <tr> <td>Application Received (Date)</td> <td>Program Officer</td> </tr> <tr> <td colspan="2">Company Name</td> </tr> <tr> <td colspan="2">Legal Name (if different)</td> </tr> <tr> <td>Mailing Address</td> <td>City/Address</td> </tr> <tr> <td colspan="2">Contact Person</td> </tr> <tr> <td>Name</td> <td></td> </tr> <tr> <td>Title</td> <td></td> </tr> <tr> <td>Phone</td> <td>Fax</td> <td>Email</td> <td>Website</td> </tr> <tr> <td colspan="4">Business Information</td> </tr> <tr> <td>Business Number (INN)</td> <td>Business Sector</td> <td>Type of Organization</td> <td></td> </tr> <tr> <td>In business since</td> <td colspan="3"> <input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other </td> </tr> <tr> <td>Total Annual Sales</td> <td colspan="3"></td> </tr> <tr> <td>Total Annual Exports</td> <td colspan="3"></td> </tr> <tr> <td># of PEI Employees</td> <td>Full-time</td> <td>Part-time</td> <td> <input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify) </td> </tr> <tr> <td colspan="4">INNOVATION PEI Business Development Application Form - 2</td> </tr> </table>		Application Received (Date)	Program Officer	Company Name		Legal Name (if different)		Mailing Address	City/Address	Contact Person		Name		Title		Phone	Fax	Email	Website	Business Information				Business Number (INN)	Business Sector	Type of Organization		In business since	<input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other			Total Annual Sales				Total Annual Exports				# of PEI Employees	Full-time	Part-time	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)	INNOVATION PEI Business Development Application Form - 2			
Application Received (Date)	Program Officer																																																																																														
Company Name																																																																																															
Legal Name (if different)																																																																																															
Mailing Address	City/Address																																																																																														
Contact Person																																																																																															
Name																																																																																															
Title																																																																																															
Phone	Fax	Email	Website																																																																																												
Business Information																																																																																															
Business Number (INN)	Business Sector	Type of Organization																																																																																													
In business since	<input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other																																																																																														
Total Annual Sales																																																																																															
Total Annual Exports																																																																																															
# of PEI Employees	Full-time	Part-time	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)																																																																																												
INNOVATION PEI Business Development Application Form - 2																																																																																															
Application Received (Date)	Program Officer																																																																																														
Company Name																																																																																															
Legal Name (if different)																																																																																															
Mailing Address	City/Address																																																																																														
Contact Person																																																																																															
Name																																																																																															
Title																																																																																															
Phone	Fax	Email	Website																																																																																												
Business Information																																																																																															
Business Number (INN)	Business Sector	Type of Organization																																																																																													
In business since	<input type="checkbox"/> Nonprofit <input type="checkbox"/> For Profit <input type="checkbox"/> Information Technology <input type="checkbox"/> Renewable Energy <input type="checkbox"/> Advanced Manufacturing <input type="checkbox"/> Craft & Software <input type="checkbox"/> Food Development <input type="checkbox"/> Services <input checked="" type="checkbox"/> Other																																																																																														
Total Annual Sales																																																																																															
Total Annual Exports																																																																																															
# of PEI Employees	Full-time	Part-time	<input type="checkbox"/> Incorporated <input type="checkbox"/> To be incorporated <input type="checkbox"/> Federally incorporated <input type="checkbox"/> Partnership <input type="checkbox"/> Sole Proprietorship <input type="checkbox"/> Cooperative <input type="checkbox"/> Other (Specify)																																																																																												
INNOVATION PEI Business Development Application Form - 2																																																																																															

Figure 123. Image Operations: Despeckle

Horizontal/Vertical line removal

Removes horizontal and/or vertical lines from the image and produces a binary (black & white) image.



Figure 124. Image Operations: Horizontal/Vertical line removal

Remove punch holes

Removes black punch holes from the image.

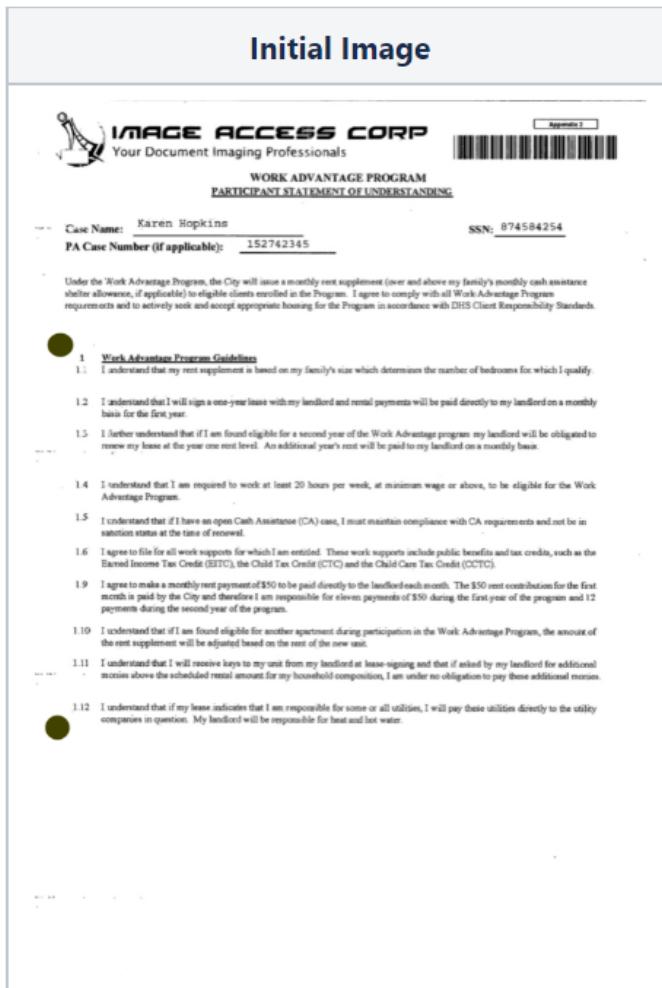
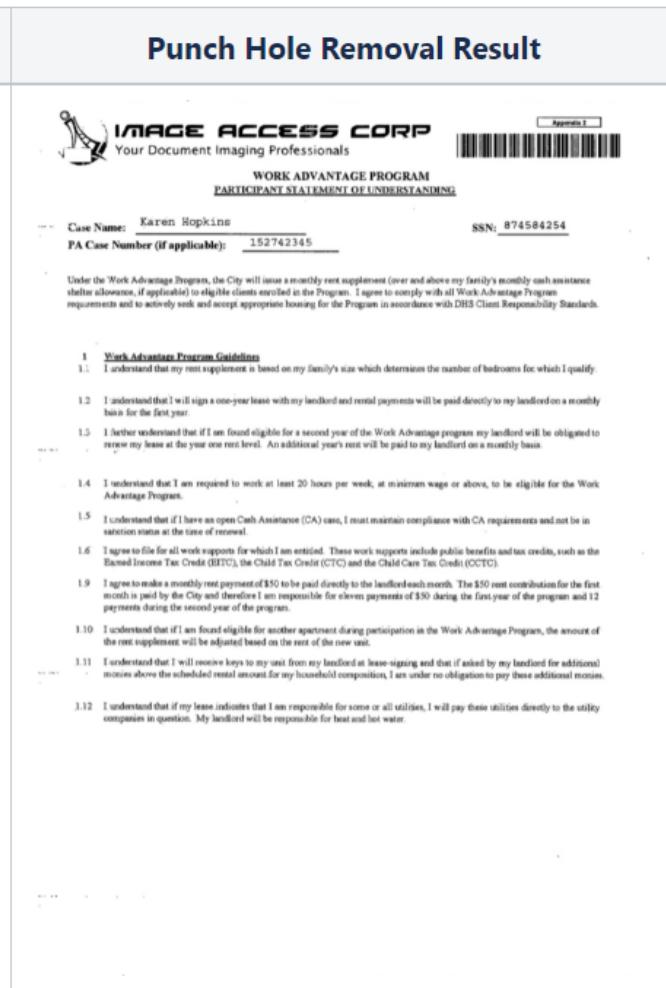
Initial Image	Punch Hole Removal Result
 <p>IMAGE ACCESS CORP Your Document Imaging Professionals</p> <p>WORK ADVANTAGE PROGRAM PARTICIPANT STATEMENT OF UNDERSTANDING</p> <p>Case Name: Karen Hopkins PA Case Number (if applicable): 152742345</p> <p>SSN: 874584254</p> <p>Under the Work Advantage Program, the City will issue a monthly rent supplement (over and above my family's monthly cash assistance shelter allowance, if applicable) to eligible clients enrolled in the Program. I agree to comply with all Work Advantage Program requirements and to actively seek and accept appropriate housing for the Program in accordance with DHS Client Responsibility Standards.</p> <p>1. Work Advantage Program Guidelines</p> <ol style="list-style-type: none"> 1. I understand that my rent supplement is based on my family's size which determines the number of bedrooms for which I qualify. 2. I understand that I will sign a one-year lease with my landlord and rental payments will be paid directly to my landlord on a monthly basis for the first year. 3. I further understand that if I am found eligible for a second year of the Work Advantage program my landlord will be obligated to renew my lease at the year one rent level. An additional year's rent will be paid to my landlord on a monthly basis. 4. I understand that I am required to work at least 20 hours per week, at minimum wage or above, to be eligible for the Work Advantage Program. 5. I understand that if I have an open Cash Assistance (CA) case, I must maintain compliance with CA requirements and not be in sanction status at the time of renewal. 6. I agree to file for all work supports for which I am entitled. These work supports include public benefits and tax credits, such as the Earned Income Tax Credit (EITC), the Child Tax Credit (CTC) and the Child Care Tax Credit (CCTC). 7. I agree to make a monthly rent payment of \$50 to be paid directly to the landlord each month. The \$50 rent contribution for the first month is paid by the City and therefore I am responsible for eleven payments of \$50 during the first year of the program and 12 payments during the second year of the program. 8. I understand that if I am found eligible for another apartment during participation in the Work Advantage Program, the amount of the rent supplement will be adjusted based on the rent of the new unit. 9. I understand that I will receive keys to my unit from my landlord at lease-signing and that if asked by my landlord for additional money above the scheduled rental amount for my household composition, I am under no obligation to pay these additional monies. 10. I understand that if my lease indicates that I am responsible for some or all utilities, I will pay these utilities directly to the utility companies in question. My landlord will be responsible for heat and hot water. 	 <p>IMAGE ACCESS CORP Your Document Imaging Professionals</p> <p>WORK ADVANTAGE PROGRAM PARTICIPANT STATEMENT OF UNDERSTANDING</p> <p>Case Name: Karen Hopkins PA Case Number (if applicable): 152742345</p> <p>SSN: 874584254</p> <p>Under the Work Advantage Program, the City will issue a monthly rent supplement (over and above my family's monthly cash assistance shelter allowance, if applicable) to eligible clients enrolled in the Program. I agree to comply with all Work Advantage Program requirements and to actively seek and accept appropriate housing for the Program in accordance with DHS Client Responsibility Standards.</p> <p>1. Work Advantage Program Guidelines</p> <ol style="list-style-type: none"> 1. I understand that my rent supplement is based on my family's size which determines the number of bedrooms for which I qualify. 2. I understand that I will sign a one-year lease with my landlord and rental payments will be paid directly to my landlord on a monthly basis for the first year. 3. I further understand that if I am found eligible for a second year of the Work Advantage program my landlord will be obligated to renew my lease at the year one rent level. An additional year's rent will be paid to my landlord on a monthly basis. 4. I understand that I am required to work at least 20 hours per week, at minimum wage or above, to be eligible for the Work Advantage Program. 5. I understand that if I have an open Cash Assistance (CA) case, I must maintain compliance with CA requirements and not be in sanction status at the time of renewal. 6. I agree to file for all work supports for which I am entitled. These work supports include public benefits and tax credits, such as the Earned Income Tax Credit (EITC), the Child Tax Credit (CTC) and the Child Care Tax Credit (CCTC). 7. I agree to make a monthly rent payment of \$50 to be paid directly to the landlord each month. The \$50 rent contribution for the first month is paid by the City and therefore I am responsible for eleven payments of \$50 during the first year of the program and 12 payments during the second year of the program. 8. I understand that if I am found eligible for another apartment during participation in the Work Advantage Program, the amount of the rent supplement will be adjusted based on the rent of the new unit. 9. I understand that I will receive keys to my unit from my landlord at lease-signing and that if asked by my landlord for additional money above the scheduled rental amount for my household composition, I am under no obligation to pay these additional monies. 10. I understand that if my lease indicates that I am responsible for some or all utilities, I will pay these utilities directly to the utility companies in question. My landlord will be responsible for heat and hot water.

Figure 125. Image Operations: Remove punch holes

3.2.5. Extraction Profiles

Info Input Solution can detect barcodes or text (using an OCR engine) while you are scanning your documents. An *Extraction Profile* contains the barcode types or language library that Info Input Solution should be looking for on a scanned page.

You can manage *Extraction Profiles* from the *Setup data dialog*, at the *Extraction Profiles tab*:

Setup data

– □ ×

Jobs Document Classes Folder Classes Field Types Extraction Profiles Data Sources Scan Profiles Export Destinations

 Filter Extraction Profiles

Name	Engine	Details	Used in #
All Barcodes	Default Barcode Reader	Codabar, Code 11, Code 128, C...	0
EAN Barcodes	Default Barcode Reader	EAN 13, EAN 8	0
PDF 417 Barcode	Default Barcode Reader	PDF 417	0

Total Extraction Profiles: 3

Close

Figure 126. Setup data dialog: Extraction Profiles tab

The *Extraction Profile properties dialog* is used to create/edit an *Extraction Profile*:

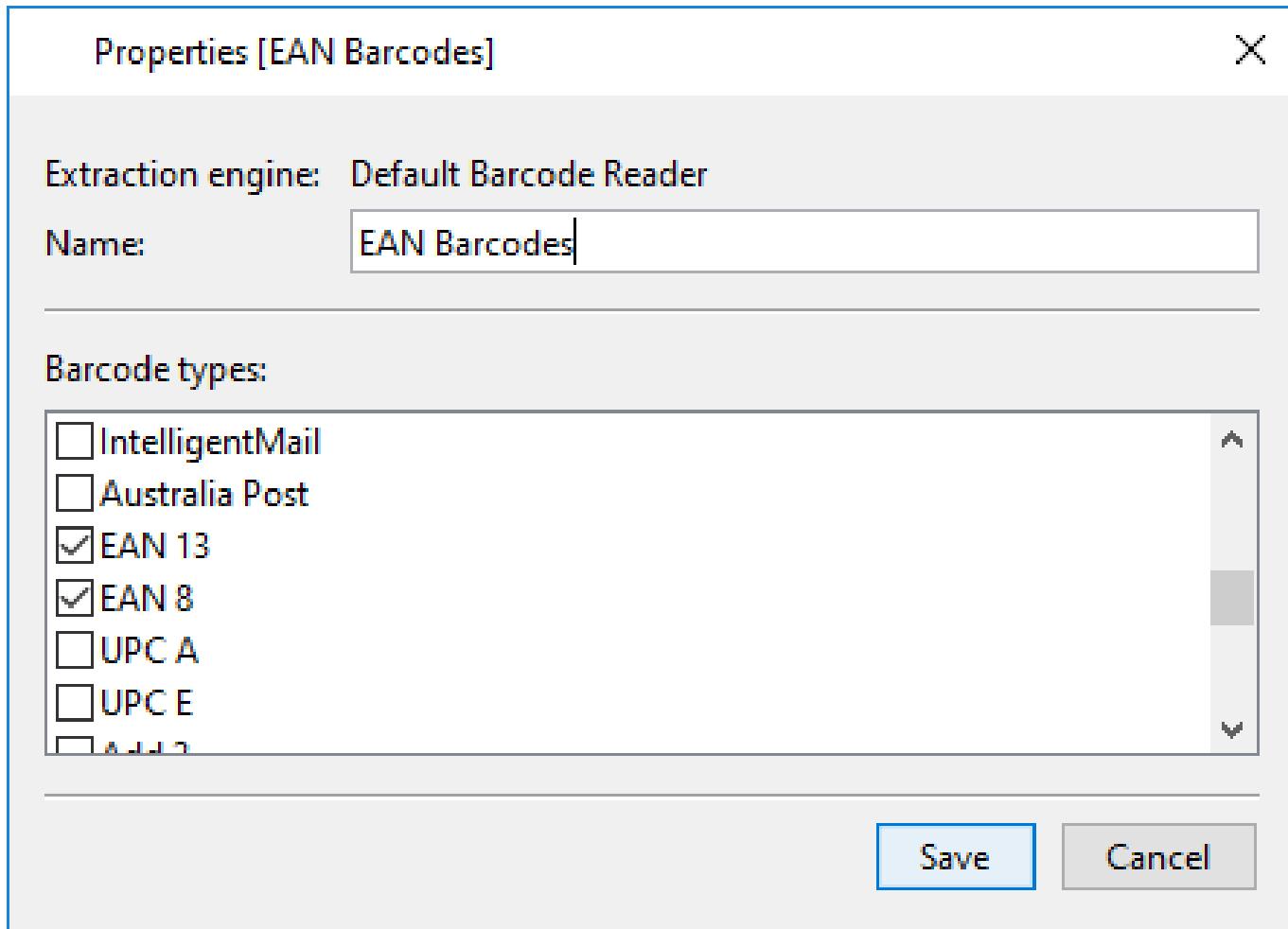


Figure 127. Extraction Profile properties dialog

You must assign a *Name* to your *Extraction Profile* and check the barcode types that you want to include in this *Extraction Profile*.

Barcode extraction is a "cpu-expensive" operation, so you should be careful to check only the barcode types that you know that exist on the types of documents you will be scanning. Selecting many barcode types on an *Extraction Profile* will decrease the performance of the Client during scanning.

An *Extraction Profile* can be assigned to a [Form Type](#). A Form Type represents a specific form of a type of document, for example an invoice form for company-A. Typically, if an invoice form has a barcode, it will be a barcode of a specific type. You should avoid assigning barcode types that are known to not exist on the form.

3.2.6. Datasources

For any operation that involves a remote database, a connection to that database needs to be established. The details of this connection are enclosed in a *Datasource*. *Datasources* are global and are

always shared by all objects that use/reference them. Moreover, unlike other shared objects, *Data-sources* are not copied (e.g. made private) when a *Job* is published: this means that changes to a *Data-source* affect all *Jobs* of the system instantly.

You can manage *Datasources* from the *Setup data dialog*, at the *Datasources tab*:

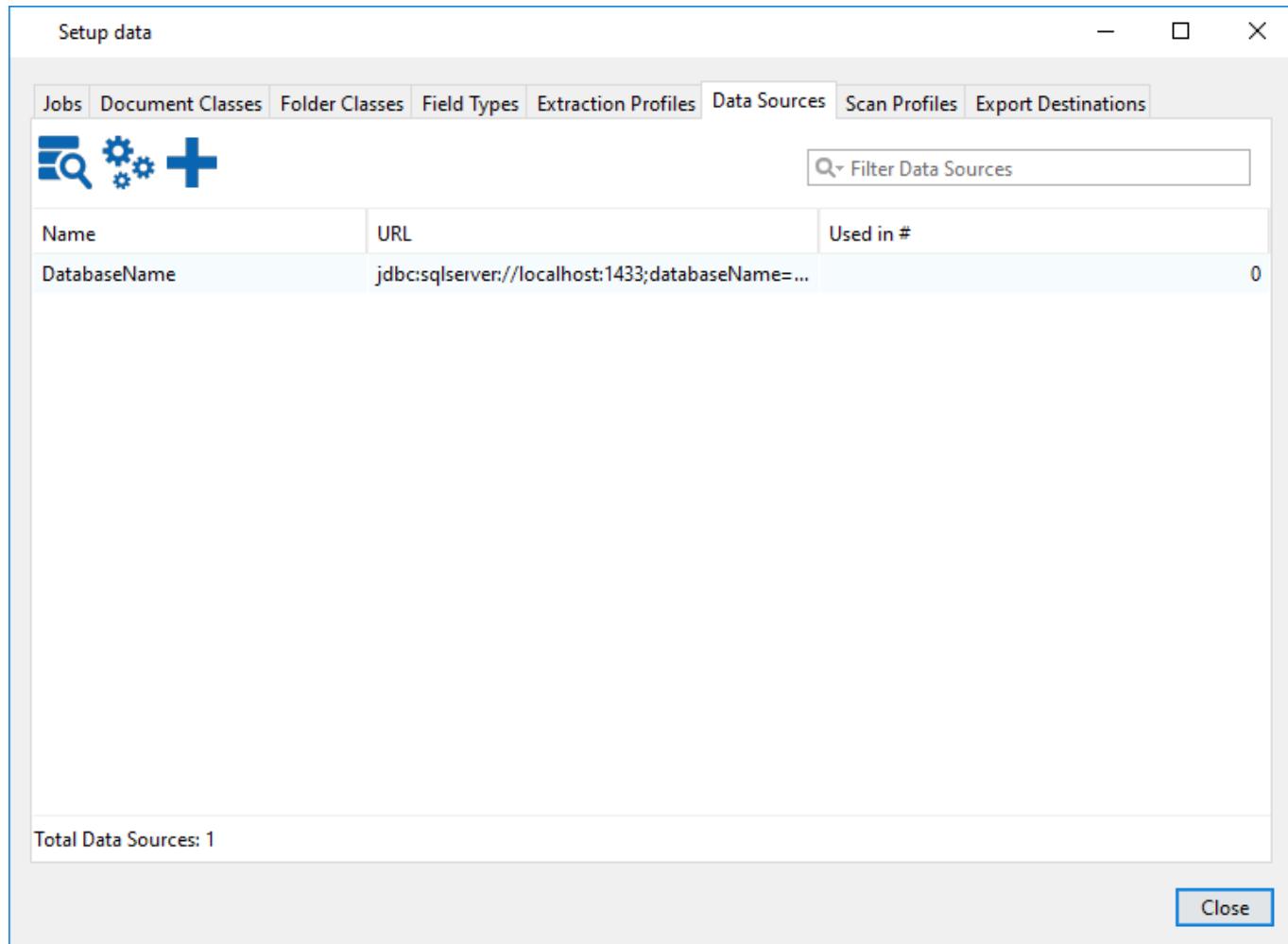


Figure 128. *Setup data dialog: Datasources tab*

The *Datasource properties dialog* is used to create/edit a *Datasource*:

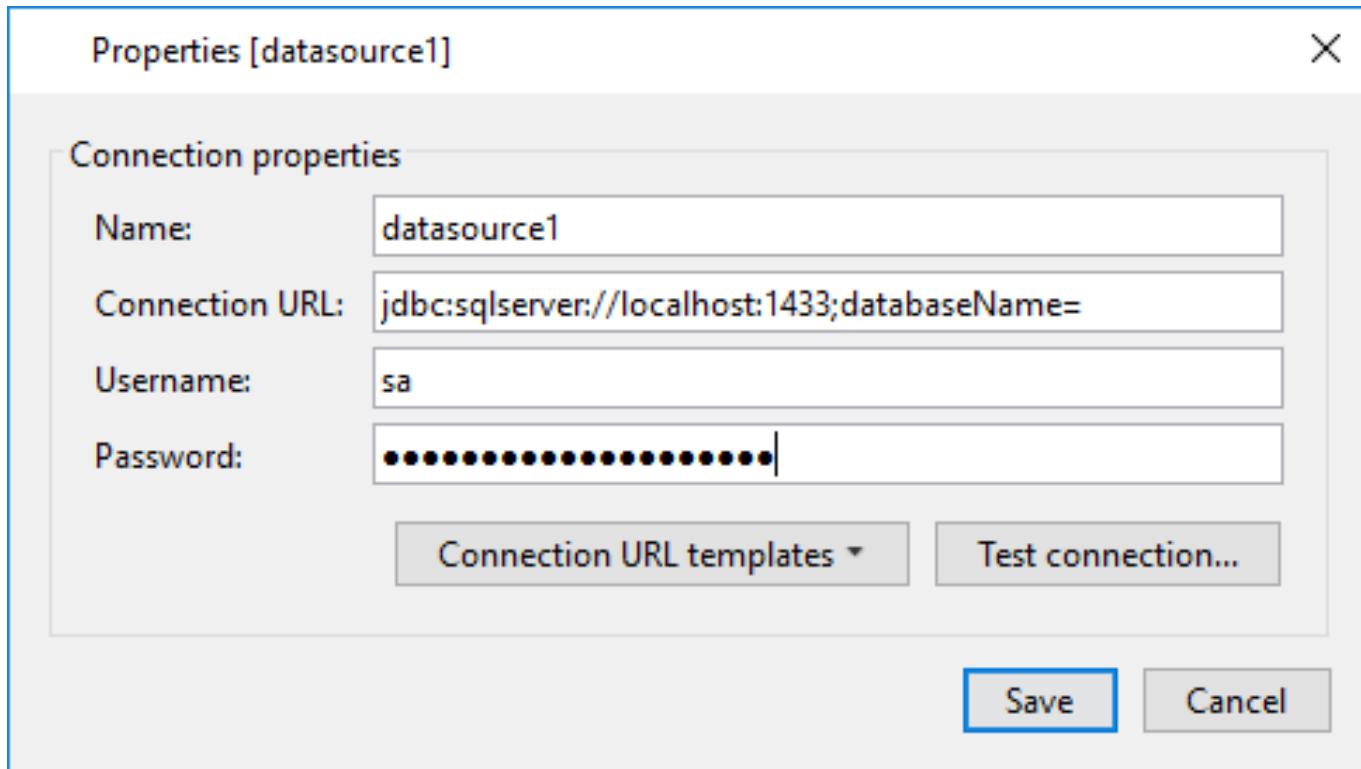


Figure 129. Datasource properties dialog

Info Input Solution uses JDBC to connect to databases and ships with drivers for MSSQL Server and Oracle Database. To create a *Datasource* the JDBC driver of the target database needs to be available on the system. Follow the steps below to add a JDBC for a database other than MSSQL or Oracle.

1. Download the appropriate JDBC driver of the database,
2. Copy it to the directory <Info Input Solution Install directory>/server/WEB-INF/lib/ of the server,
3. Then the driver name should be added in the file <Info Input Solution Install directory>\server\WEB-INF\web.xml in the property `scanserver.util.JdbcDriverExplicitLoader`. See the XML section below,

```
<listener>
  <listener-class>scanserver.util.JdbcDriverExplicitLoader</listener-class>
</listener>
<context-param>
  <param-name>jdbc.drivers</param-name>
  <param-value>
    [type the full driver class name here]
  </param-value>
</context-param>
```

```
</context-param>
```

4. The Core Service needs to be restarted for the changes to take place.

Then from the *Job Setup* select *Datasources* and press the plus button to create a new *Datasource* you need to fill in the *Name*, *Connection URL*, *Username* and *Password* fields in the definition dialog. The *Connection URL templates* button provides the expected *JDBC URLs* for some common databases: you may select one to pre-populate the *Connection URL text box* and then update the placeholder values with the real ones (notice that some JDBC drivers may require a slightly different URL: please refer to the JDBC driver documentation for the exact format). After you have filled in all four pieces of information, click the *Test connection...* button to test the connectivity to the database.

3.3. Scripting

The Thick Client comes with embedded Javascript engine, and the HTML Client runs from within the browser. Therefore, The functionality of both the Thick Client and the HTML Client can be enhanced by adding additional script in the Job configuration. Description about the scripting context and available operations is outside of this guide's scope and can be found in the Info Input Solution Developers Guide. Therefore, this guide will provide a summary about the scripting context that is available.

3.3.1. Scripting Levels

For both {CL_JAVA_HTML}s additional Javascript scripting can be used to modify and/or extend certain functionality of the system.

Scripts can be defined at 4 levels:

- Global script
- Job level scripts
- Index class level scripts (Batch, Folder and Document Level)
- Workflow Step scripts (server-side script steps and client-side and branch steps)

Details and examples about the events, that are supported for each script, can be found within the scripts themselves. When you select to edit a script, a default script template that contains functions with empty bodies is presented. You may use this script template to write your code and extend it accordingly.

3.3.1.1. Global script

There is a single *global script* that applies to the whole system (not just the Client). It can be accessed from the *Tools & Options menu* → *Global script...* item. The global script allows you to:

- enforce a password verification scheme in order to setup specific requirements for user passwords.
- modify the *New Batch dialog* behavior.
- setup a *default batch* to create whenever a user drag-n-drops files on the Client) and no batch exists.

What is more important to remember is that, the global script is *always included* in other scripts. So this is the best place to put functions that are global in nature and may be called by other scripts. For example, if you want to write an SSN validation function and re-use it from multiple index class level scripts, you can code it in the global script and just call it from other scripts.

3.3.1.2. Job level scripts

You define job level scripts by selecting *Edit Script* from the [General Options](#). Job level scripts apply to a specific job.

3.3.1.3. Index class level scripts

See section on [Indexing Scripting](#).

3.3.1.4. Workflow Step scripts

See section on [Workflow](#).

3.3.2. Variables

Info Input Solution supports the use of *Variables* within *expressions* in several places in the configuration interface. A variable in Info Input Solution may represent a global value, like the *current date* or *time*, or the property of an object, like the *name* or *description* of a document or the value of an *index field*.

The syntax of a variable is: `${[prefix].[name][:arguments]}`. The curly brackets may be omitted if there are no *spaces* or *prefix* or *arguments*.

Examples of valid variables are: `$CurrentDate`, `$UUID`, `${CurrentDate}`, `${idx.Address}`, `${parent.parent.name}`, `${BarCode:1:3:1}`, etc.

3.3.2.1. Standard Variables

The following table lists the *standard variables*. Notice that the value of a variable depends on the context, and specifically the node it is applied on. For example, the `ChildCount` variable always returns the number of child nodes of a *node* and applies for all levels (`batch`, `folder`, `document`). The `JobName` variable also applies to all *nodes*, even if it is evaluated for a `document` (since a `document` always is part of a `batch` that belongs to a `job`). The `FolderID` for example has a *value* only if evaluated for a `folder`, `document` or `page` *node*, but not for a `batch` *node*.

Variable name	Explanation
ID	Returns the internal numeric ID of the node (unique for the system). If the batch has not been committed to the Core Service, this value is 0.
ParentID	The ID of the parent of the node
Index	The index of this node, relative to its parent (1-based)
AbsoluteIndex	The index of this node for this level (e.g. if this is a page node, then this is the index of the page in the whole batch, ignoring the document containment and hierarchy). It is 1-based.
LevelName	The name of the Level of the current node. The default values are Job , Batch , Folder , Document , Page .
ChildCount	The number of immediate children for this node
JobID	The ID of the Job
JobName	The name of the Job
BatchID	the ID of the Batch
BatchName	the name of the Batch
BatchDescription	the description of the Batch
BatchCreationDate	the Creation Date of the Batch
BatchCreationTime	the Creation Time of the Batch
BatchPath	the local temporary path where the Batch is stored
FolderID	the ID of the Folder
FolderIndex	the Index of the Folder
FolderAbsoluteIndex	the Absolute Index of the Folder
FolderCount	the number of Folders for this Batch
FolderClassName	the assigned Folder Class Name for this folder
DocumentID	the ID of the Document
DocumentIndex	the Index of the Document
DocumentAbsoluteIndex	the Absolute Index of the Document
DocumentCount	the number of Documents for this node (if a Batch that has folder, returns the sum of all documents for all folders)
DocumentClassName	the Document Class Name for this document
DocumentFormName	the form type name of this document

Variable name	Explanation
OriginalFileName	the original file name for this document, if this is an eDocument
FirstPageOriginalFileName	the original file name of the first page of the document
PageCount	the number of Page for this node (if this is a folder or batch, returns the sum for all pages below)
PageIndex	the Index of this Page
PageAbsoluteIndex	the Absolute Index of this Page
BarCode	Applies to a document object only. It may take 0, 1 or 2 arguments. The first argument is the index of the barcode in the page to return, default is 1. The second argument is the page of the document to return the barcode from, default value is 1. For example, the variable <code> \${BarCode:2:3}</code> will return the second barcode of the 3rd page of the document.
Counter	This is a global counter that returns the next available integer (starting from 1) each time it is evaluated. The numbers are evaluated centrally for all users. It may take 1 argument that is the name (domain) of the counter: this allows you to create multiple distinct counters. For example <code> \${Counter:bnumbers}</code> defines a counter named <code>bnumbers</code> that returns 1, 2, 3, ... each time it is evaluated. There is no provision to reset a counter.
CurrentDate	returns the current date. It may take one argument that is the format of the date. For supported format see the field types topic and the available formats for the <code>DATE</code> data type. For example the variable <code> \${Current-Date:MMM/yyyy}</code> returns <code>October/2013</code> if evaluated for the month of <code>October</code> and the year <code>2013</code> .
CurrentTime	returns the current time. It may take one argument similar to the <code>Current-Date</code> variable (see above).
OSComputerName	the name of the user's computer
OSUserName	the operating system user name
NetBiosName	the netbios computer name
OperatorUsername	the user's username
OperatorFullname	The user's full name
OperatorDescription	The user's description
UTCOffset	Offset from UTC time, not including daylight savings offset
DSTOffset	Daylight savings offset

Variable name	Explanation
UTCOffset	Offset from UTC time including daylight savings offset
GUID	globally unique ID for the node
Empty	an empty value
Name	the name of the node

3.3.2.2. Variable prefixes

The following prefixes are recognized in variables:

Prefix	Description
parent.	references the parent node. For example, the variable \${parent.name} when applied to a folder will return the name of the batch (which is the parent of the folder). Can be used more than once (e.g. \${parent.parent.name})
batch.	references the top level batch node. For example, the variable \${batch.name} when applied to a folder or document, will return the name of the batch.
folder.	references the folder node. For example, the variable \${folder.name} when applied to a folder or document, will return the name of the folder.
idx.	references an <i>index field value</i> . For example, the variable \${idx.name} will return the <i>value</i> of an <i>index field</i> called name (instead of the name of the current node)
property.	references an arbitrary and <i>dynamically defined property</i> of the node. If the property does not exist it is created the first time it is referenced.

3.3.2.3. Dynamically defined properties on nodes

By using the **property.** prefix, you may dynamically define arbitrary properties on nodes. Unlike **idx.** values, which are predefined variables and need to be defined beforehand during *Indexing Setup*, the dynamic properties do not require any definition but are automatically defined for a node the first time they are referenced.

Dynamic properties can be referenced and/or defined in a *script* using the **properties** property of a node. For example, the following script assigns a **scantime** property and sets it to the current time during scanning:

```
function pageScanned(batch, document, page){
  var d = new java.util.Date();
  page.properties['scantime'] = d.toString();
```

```
debug.print('scantime is ' + page.properties['scantime']);  
}
```

4. Scan Profiles Management

A Scan Profile encapsulates the information required to capture images and documents in Info Input Solution from a specific source.

Info Input Solution supports four different sources for acquiring images and documents:

- Import of images and files from a directory (local or network)
- TWAIN compatible scanners
- ISIS compatible scanners
- System Clipboard
- Capture of images from an iOS based mobile device (iPad or iPhone)

Typically a *user* creates a *Scan Profile* and sets it up in order to use a scanning device that is connected to his/her workstation. *Scan Profiles* that are created by a *user* are stored in Info Input Solution Database and they are associated with the MAC address of the workstation where they were created. These *Scan Profiles* are considered *local* for a given workstation.

Info Input Solution allows the creation of [Global Scan Profiles](#), which can be used by any *user* at any workstation. The *Administrator* can assign specific *Global Scan Profiles* to a certain *Job*, in order to control the scanning properties that will be used while scanning.

4.1. Defining Scan Profiles

From the *Tools & Options* menu, select *Manage Scan Profiles...* to view all available local *Scan Profiles* :

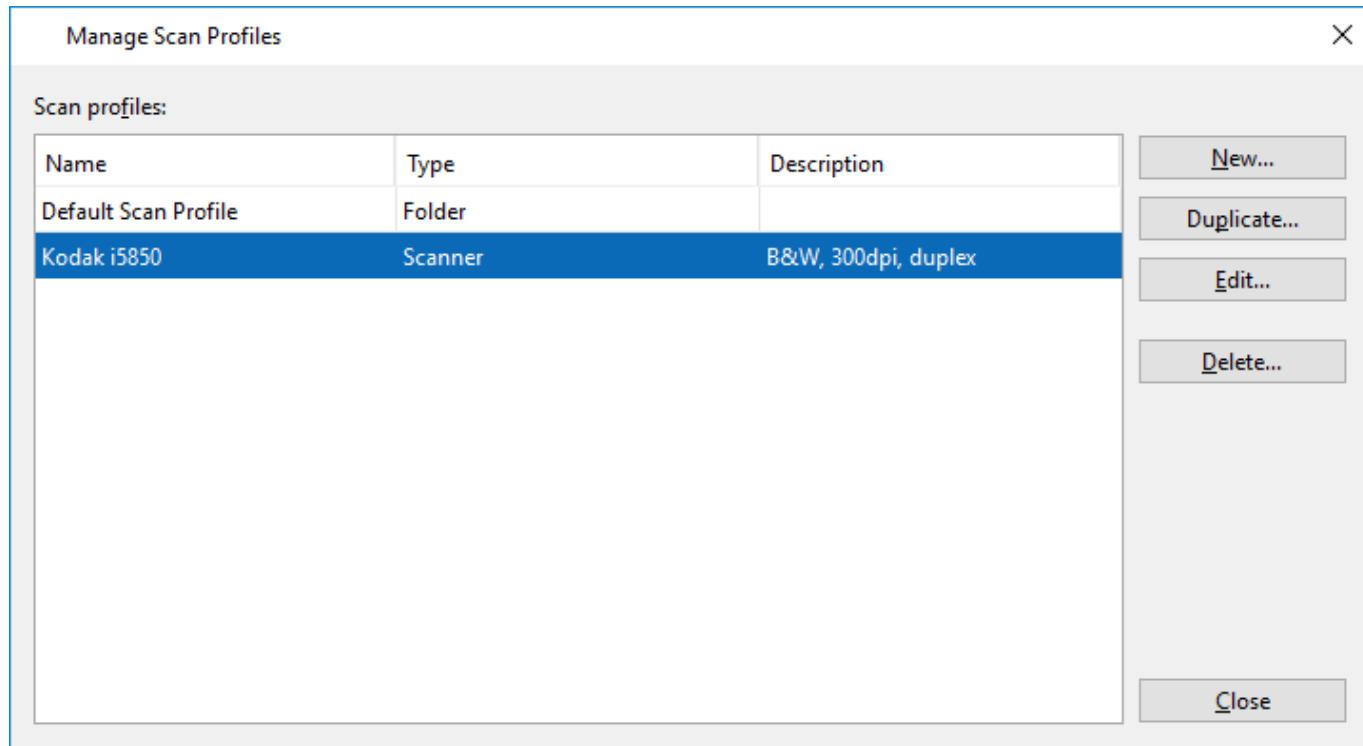


Figure 130. Manage Scan Profiles dialog

The *Manage Scan Profiles dialog* allows you to create new *Scan Profiles*, as well as *Duplicate*, *Edit* or *Delete* existing *Scan Profiles*.

Note that if you delete all *Scan Profiles*, a default *Folder profile* will be created automatically.

Click *New...* to create a new *Scan Profile*. The *Define Scan Profile dialog* appears:

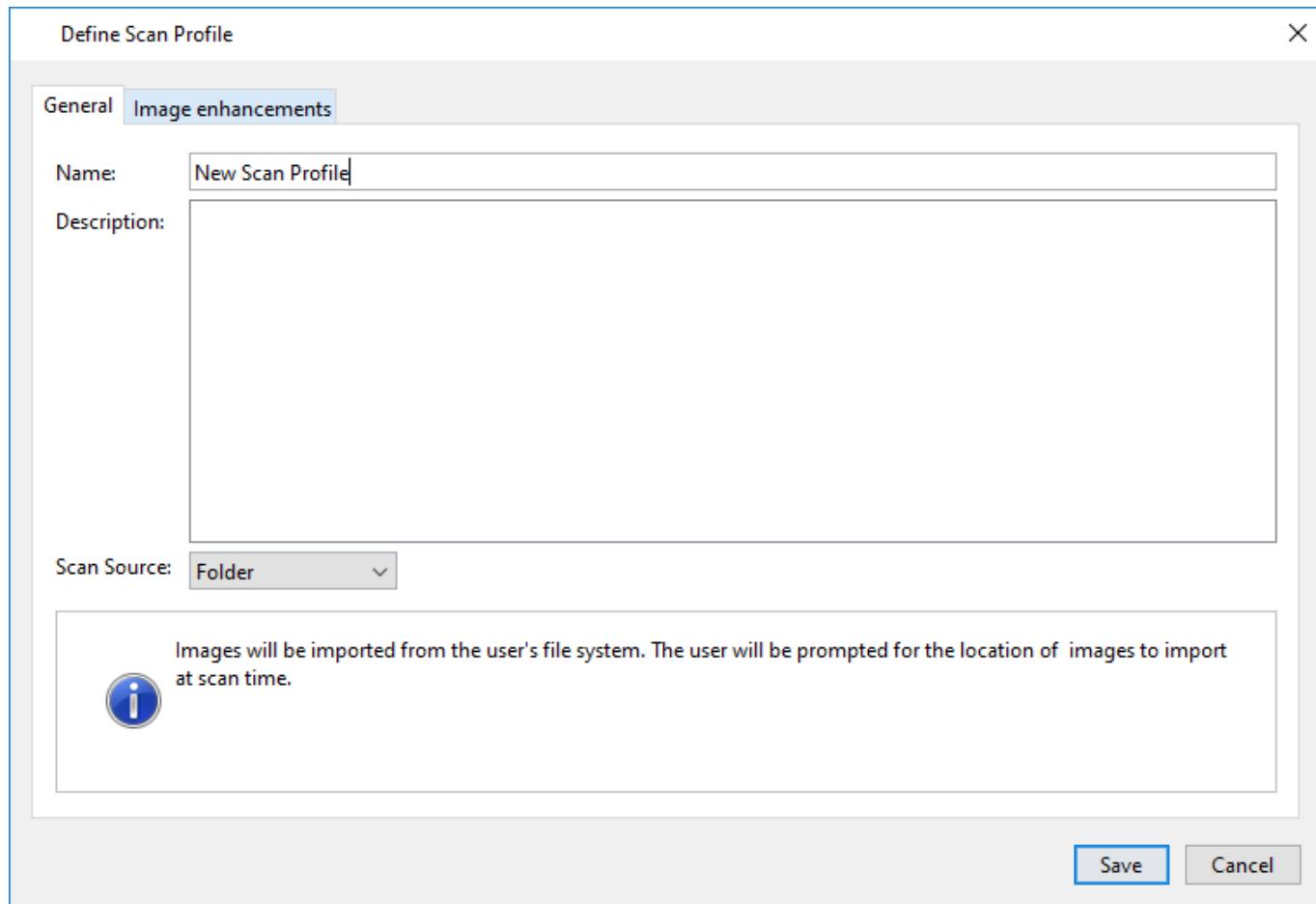


Figure 131. Define Scan Profile dialog

Write a *Name* and short *Description* for the new *Scan Profile*. Select the *type* of source for this *Scan Profile*, from the *Scan Source* combo box. Info Input Solution supports the following four kinds of *Scan Sources*:

Folder

Use this option to create a *Scan Profile* that allows you to import images and documents from local or network file system.

Scanner

Use this option for TWAIN compatible scanners. From the *Scanner list* select the scanner you want to use. Notice that this list only displays the scanners for which you have installed their drivers. Click on the *Setup* button to open the scanner's native setup dialog to select the scan parameters for this profile. Info Input Solution utilizes the scanner's native setup dialog to give you access to all the different parameters and functions for each scanner, saves them in the *Scan Profile* and applies them during scanning: thus you can parameterize the scanning up to the last detail, according to your scanners' capabilities.

Driver less Network Scanning

Info Input Solution provides the ability to Scan through the HTTP/HTTPS protocol by connecting to scanners on the same Network(Only available through the HTML Client).

OpenText Pixtools for web

Info Input Solution provides tight integration with the *EMC Captiva Cloud toolkit* which allows you to use ISIS drivers and take advantage of their web-only scan-architecture. From the *Scan Source*, select *Pixtools for Web*: the *Scanner list* will be populated with a list of scanners that are supported by the *EMC Captiva Cloud toolkit*. Click on the *Setup* button to bring up the *setup dialog* of the *EMC Captiva Cloud toolkit* for the specific scanner.

System Clipboard

Info Input Solution also supports importing images that are temporarily placed on the system clipboard. When using this Scan Profile, the user can repeatedly select and store images on the clipboard, and these images will be automatically imported to the current batch, in the same order.

Note that you may create many *Scan Profiles* of the same type and for the same device. For example, you could have separate *Scan Profiles* for color and for Black & White scanning and use them accordingly, depending on the type of document you are scanning.

4.2. Global Scan Profiles

Users that have the *Create Global Scan Profile* permission, or the *Job Administrator* permission (see [User and Group Administration](#)), are allowed to create a *Scan Profile* and make it *Global*, thus making it available to all users on all workstations. By default, the *Administrator* can also create/edit *Global Scan Profiles*.

[Read about the different types of Scan Profiles supported by Info Input Solution.](#)

To create/edit *Global Scan Profiles*, go to *Tools & Options menu* → *Jobs setup...* and select the *Scan Profiles* tab:

Setup data

Jobs Document Classes Folder Classes Field Types Extraction Profiles Data Sources Scan Profiles **Scan Profiles** Export Destinations



Name	Type	Description	Used in #
clipboard	System Clipboard		1
Kodak i5850	Scanner	B&W, 300dpi, duplex	2

Total Scan Profiles: 2

Close

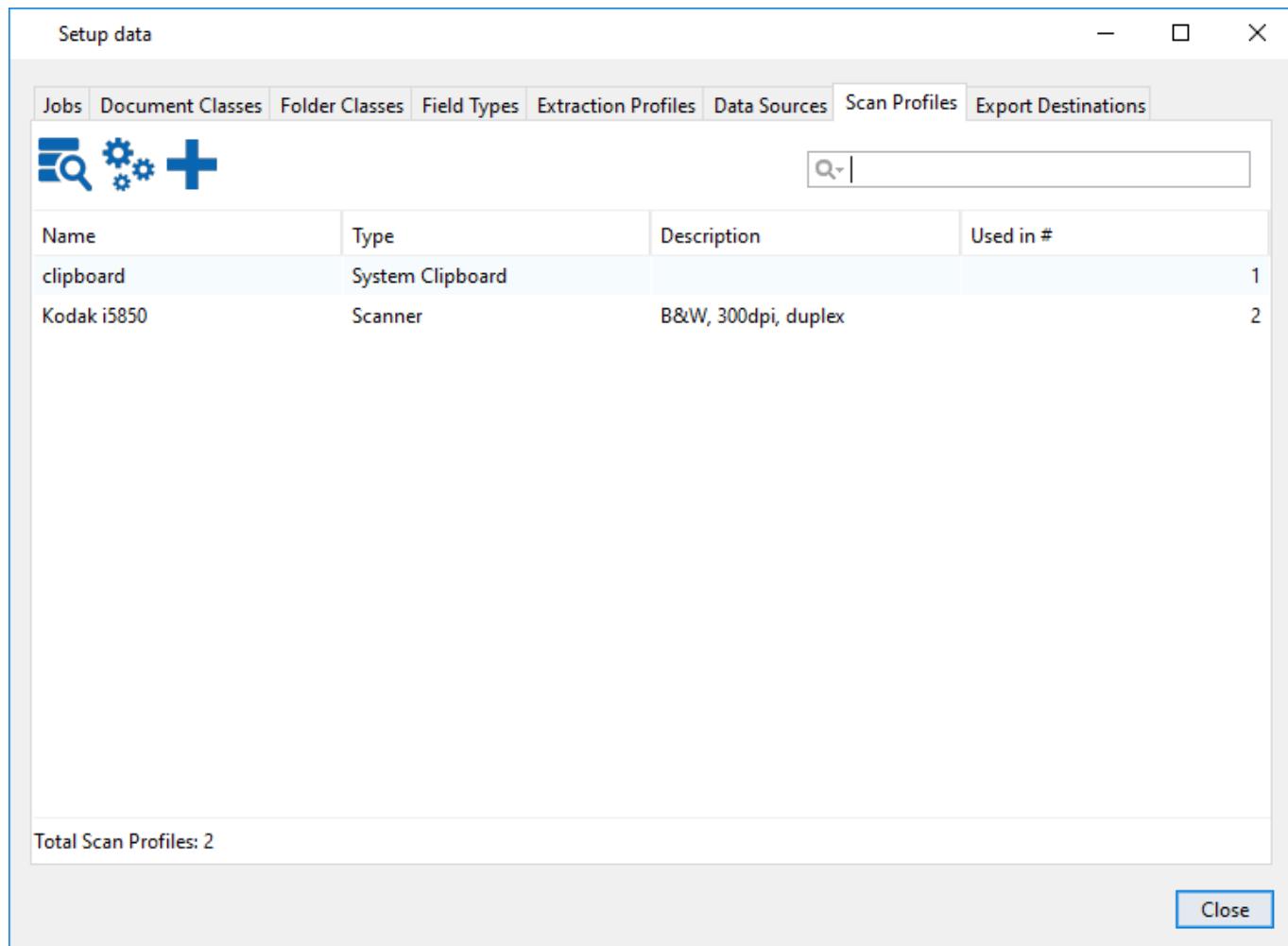


Figure 132. Setup data: Global Scan Profiles tab

Editing of *Global Scan Profiles* is done with the *Define Scan Profile dialog*:

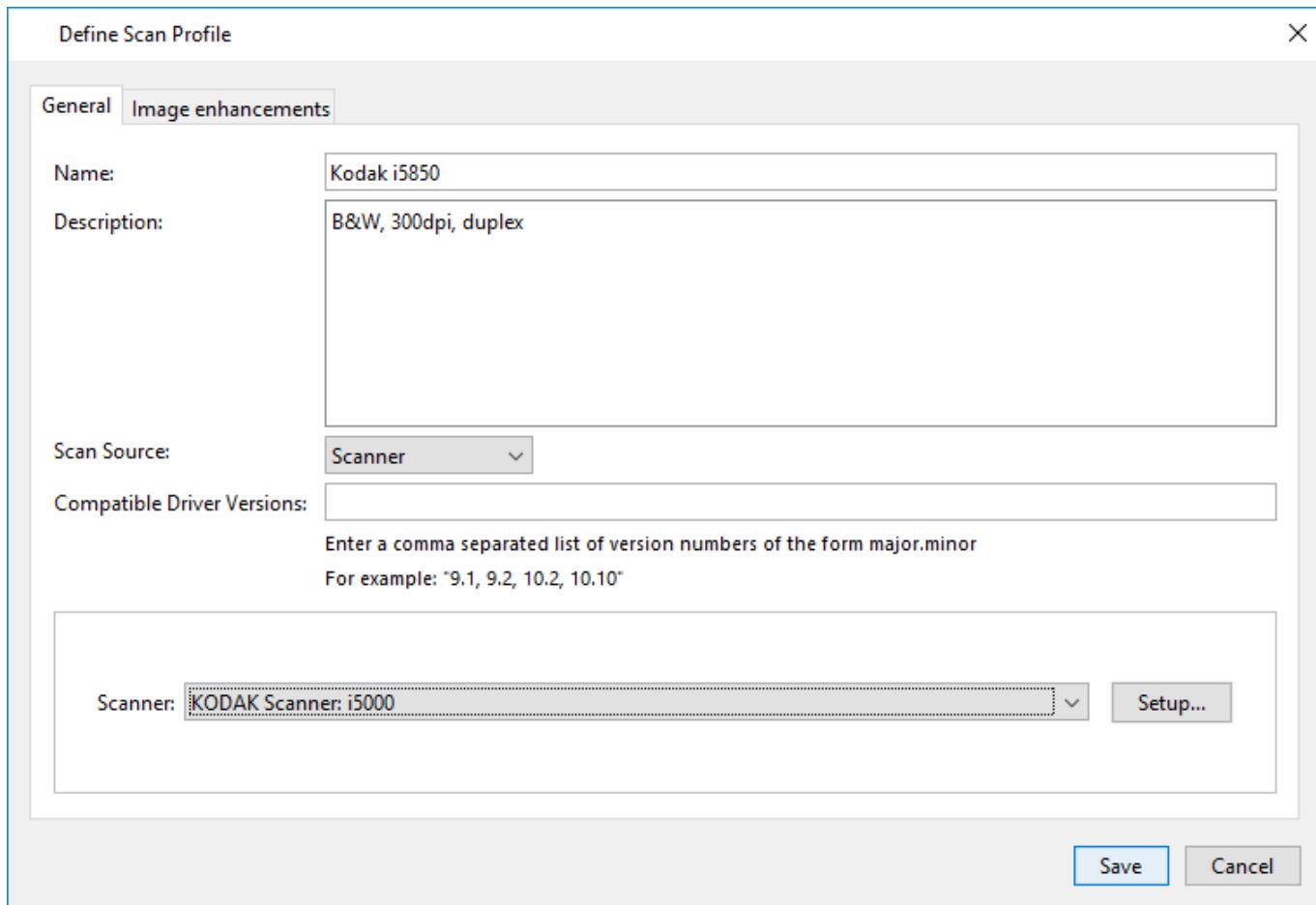


Figure 133. Dialog for creating/editing Global Scan Profile

A *Global Scan Profile* is considered to be *in use* when it has been assigned to at least one *Job*. Note that if a *Global Scan Profiles* is *in use*, then you cannot delete it. In order to do that, you need to remove it from all *Jobs* that reference it.

Next to *Compatible Driver Versions* you may enter a list of driver version numbers for which this *Scan Profile* should be used. Each driver version should be in the form `<major number>.<minor number>`, for example **9.1**, **10.10** etc. It is possible that for different driver versions, the data that Info Input Solution gets from the Scanner's native setup dialog are different. In that case, a *Scan Profile* that was created using one driver version, may not work on a workstation that has a different driver installed. You should be able to know which driver versions are installed on the user's workstation and accordingly create *Global Scan Profiles*. Info Input Solution will detect the driver version installed on a user's workstation and will try to find a match for it in the *Scan Profile's* *Compatible Driver Versions* list. If no match is found, then this *Scan Profile* may not be shown as available for scanning to the user. If a *Scan Profile* has no *Compatible Driver Versions* defined, then Info Input Solution assumes that this *Scan Profile* can work with any driver version installed.

4.3. Selecting Scan Profiles

The *active Scan Profile* is displayed on the *Status Bar*. To change the *active Scan Profile* and to modify the active scan properties, the user may use the *Select active Scan Profile dialog*:

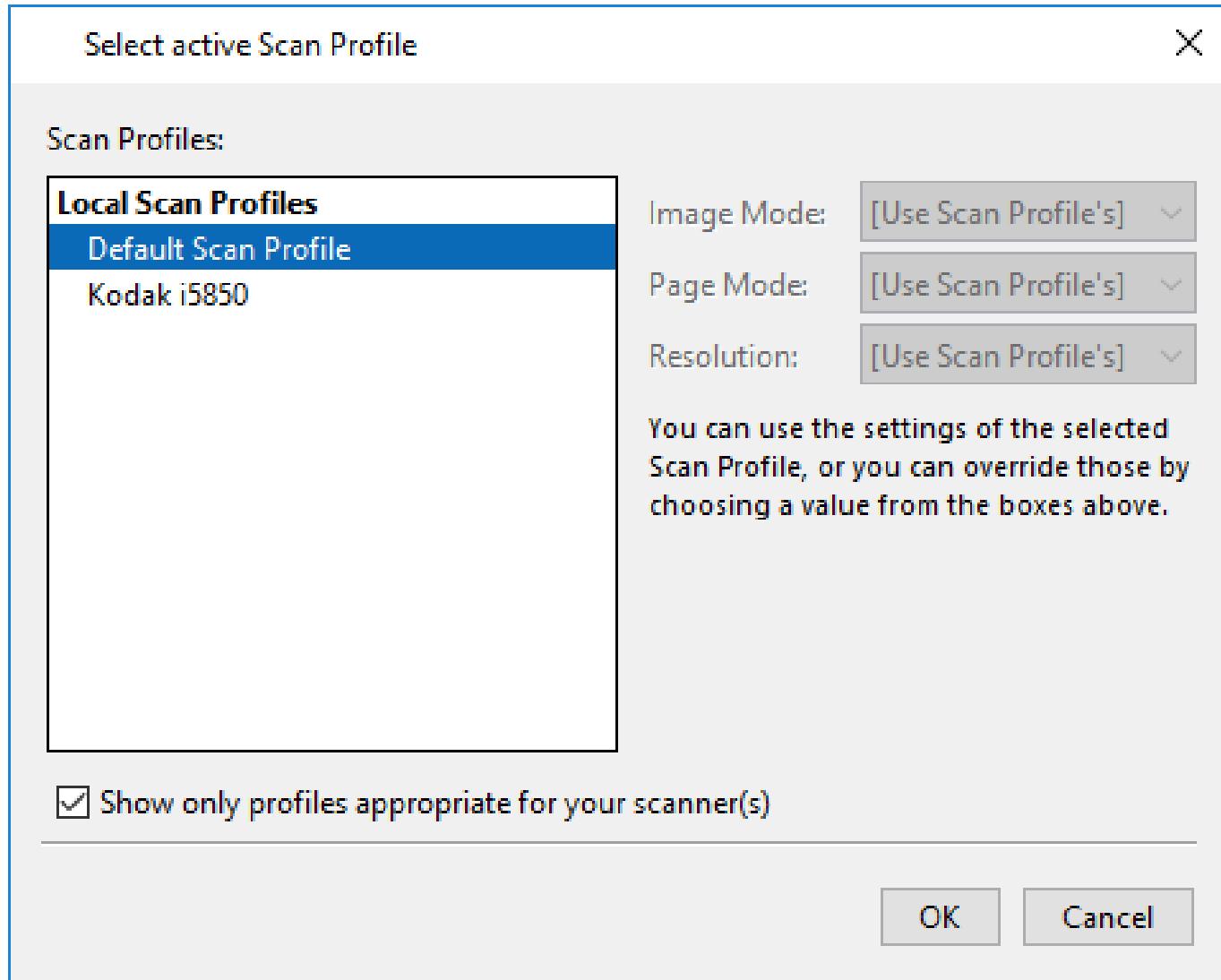


Figure 134. Select active Scan Profile dialog

This dialog can be displayed using any of the following three ways:

- By clicking on the *Active Scan Profile* label, on the *Status Bar*.
- By selecting the *Select Scan Profile...* item from the *Tools & Options menu* (≡).
- By bringing up the popup menu on the *Scan button* and selecting the *Select Scan Profile...* item. This option is available only when there is an open *Batch* in the Thick Client.

The *Scan Profiles* list displays all the *available Scan Profiles*, based on the *Job* definition for the *Batch* that

is currently open (and on the user's permissions). The list may include *local*, [Global Scan Profiles](#) or both. If there is no open Batch, then only *local Scan Profiles* will be displayed.

Info Input Solution will try to find *Scan Profiles* that are appropriate for the scanners connected to the user's workstation and for the specific driver version installed on the workstation. Typically, an organization might use several scanner models and several different drivers for the same model. So, a *Job* might have many *Global Scan Profiles* assigned. However, on each workstation only a subset of the scanners will be available and only one driver version will be installed. Info Input Solution will try to narrow down the available *Scan Profiles* by removing the ones that do not match on the workstation, in order to make the selection easier for the user. It is possible to disable this functionality by un-checking the *Show only profiles appropriate for your scanner(s)* option. In this case all the *Global Scan Profiles* assigned on the *Job* will be displayed on the list.

On the right part of the dialog, the user can select values for the *Image Mode*, *Page Mode* and *Resolution*, which will override the ones selected during the creation of the *Scan Profile*. The choices available on these combo boxes, depend on the *Scan Properties* that are assigned on the *Job* (and on the user's [permissions](#)). Please, note that certain types of *Scan Profiles* (like *Folder* and *Mobile*) do not support changes in the scan properties, and the combo boxes will be disabled when a *Scan Profile* of this type is selected.

4.4. Scan Profiles and New Batches

To create a new *Batch*, the user uses the *Create New Batch dialog*:

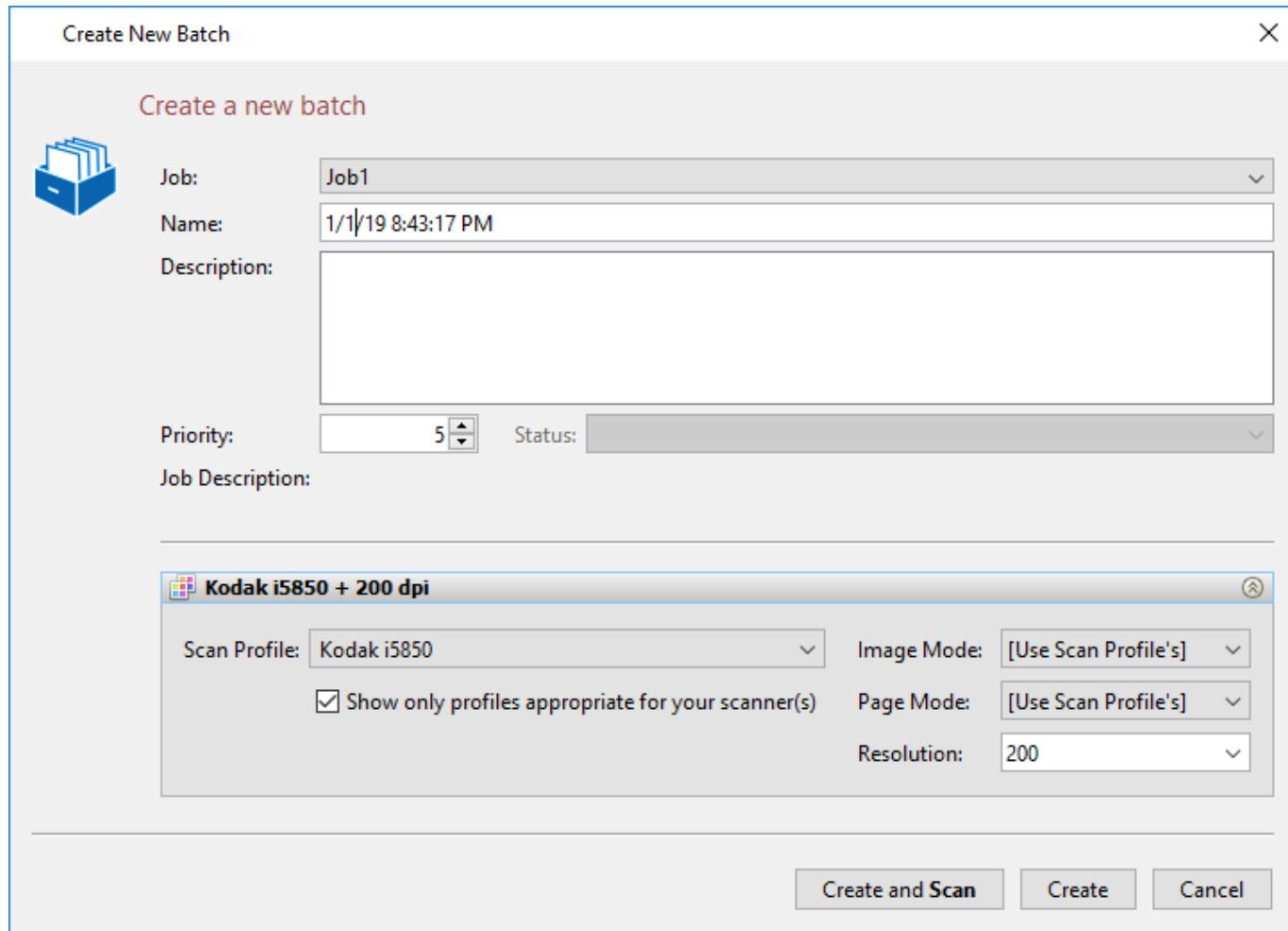


Figure 135. Create New Batch dialog with Scan Profile pane expanded

This dialog can be displayed using any of the following two ways:

- Click on the *New batch* (+) button.
- By bringing up the popup menu on the Batch button and selecting the *New batch...* item.

On the *Create New Batch* dialog, the user may select the *Job* s/he wants to use to create the new Batch and give a *name* and a *short description* to the Batch. S/he may also set certain Batch properties like the *priority*, *status* and *privacy status*, depending on the Job selected.

When a new Batch is created, the user must select which *Scan Profile* and properties to use for capturing images and documents into the Batch. The user may also change the *active Scan Profile* and properties at any time during the life cycle of the Batch. In other words, as long as the Job definition allows it, a certain Batch may contain images and documents that were scanned using different *Scan Profiles* and scan properties.

The *Create New Batch dialog* includes a collapsible pane that allows the user to select the *Scan Profile*

and properties to use for the new Batch.

The *Scan Profile* combo box includes all the allowed *Scan Profiles*, based on the selected Job's definition (and the [user's permissions](#)). The list may include *local*, [Global Scan Profiles](#) or both.

Info Input Solution will try to narrow down the available *Scan Profiles* by removing the ones that do not match on the workstation, in order to make the selection easier for the user. It is possible to disable this functionality by un-checking the *Show only profiles appropriate for your scanner(s)* option. See section [Selecting Scan Profiles](#) for more detailed explanation of this feature.

The user is also able to select the desired *Image Mode*, *Page Mode* and *Resolution* for scanning documents in the new Batch. The choices available on the corresponding combo boxes, depend on the selected Job's definition (and on the user's permissions).

Info Input Solution saves locally the user's selections for each Job that the user works on. This way, the next time a new Batch is created from the same Job, Info Input Solution will pre-select the *Scan Profile* and scan properties that were used the last time to scan documents for that Job.

5. Import Service Administration

The *Import Service* module can automatically import files from specified folders and create new batches for processing, as new files appear in the folder(s). Leveraging the same technology used by the Thick Client, a distributed *Import Service* installation can connect to the *Core Service*, upload images from local folders (or even remote locations, outside the organizations' network) and create new Batches, using a predefined Job.

5.1. Start the Import Service

During Info Input Solution installation, the *Import Service* module has been installed on the server that hosts Info Input Solution, but the required Windows Service for starting the *Import Service* is not installed by default. The *Import Service* process will consume one concurrent license, when started.

Use the Info Input Solution Administration Utility, go to Services tab and click on *Import Service* to install and start the *Import Service*. At start-up, the *Import Service* communicates and registers with the *Core Service*.

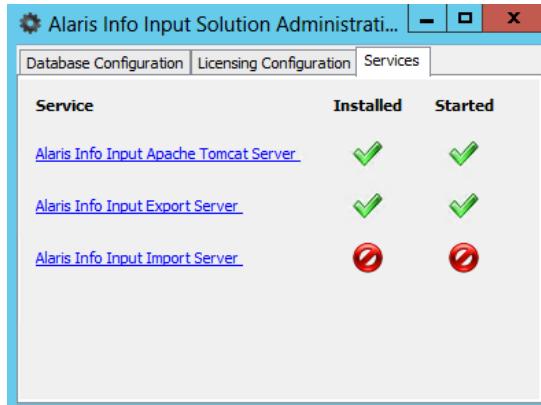
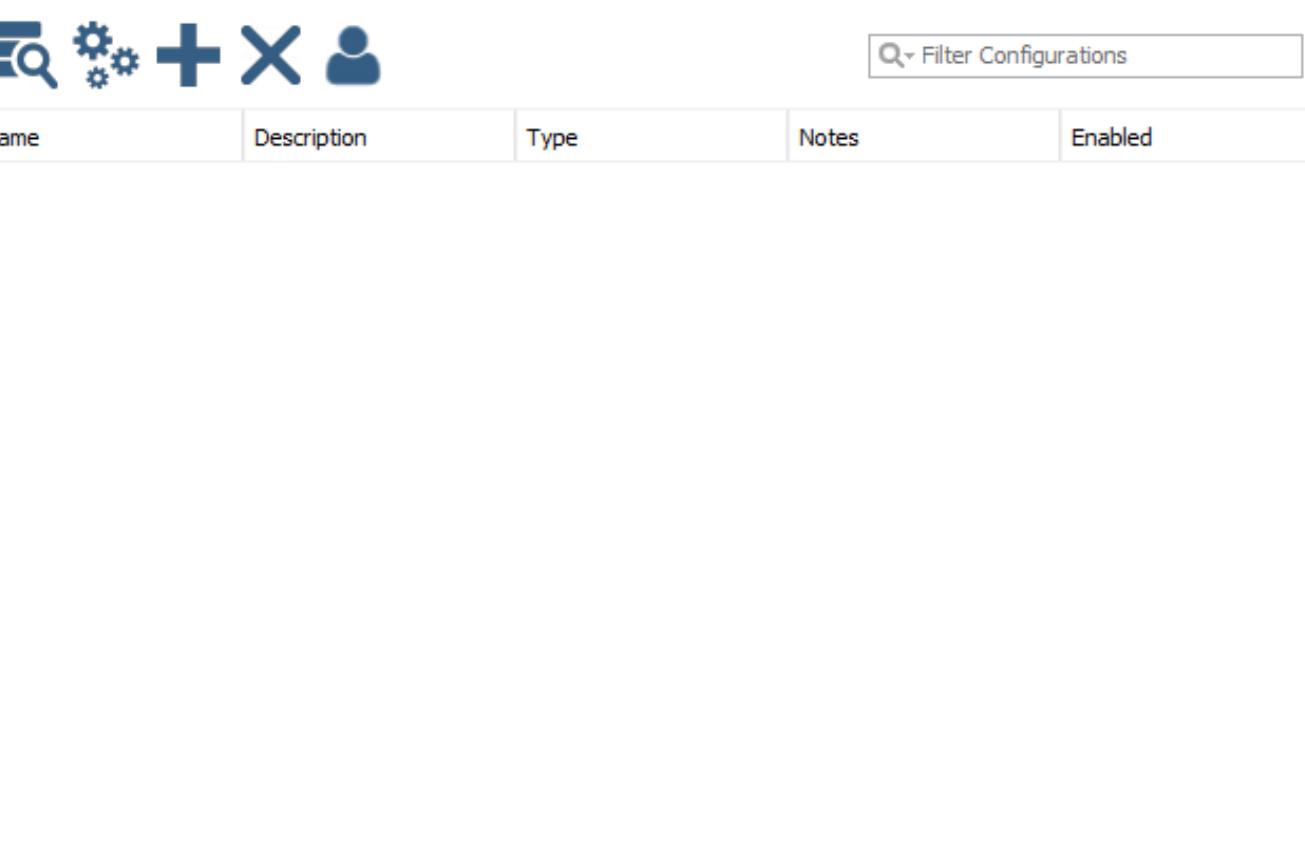


Figure 136. Install the Import Server service

5.2. Launch the Import Service Administration window

In Info Input Solution start menu, a link for launching the *Import Server Administration* window is available.

Import Server Administration



Total Configurations: 0

The  (New) button can be used to create a new Import Configuration, or select one existing Import Configuration to review the configuration properties.

The following Import configuration types are available [File System Import](#), [XML File Import](#), [Email Import](#) [Box Skill Import](#) [INFuse Import](#),

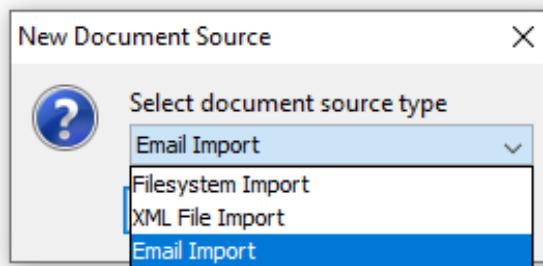


Figure 137. Import Service New Document Source



The **(Table Options)** button can be used to change the *Import Server Administration* window view options.



The **(Update Authentication)** can be used to change the User or the URL that will be used by the *Import Service* to initiate a connection with the *Core Service*.

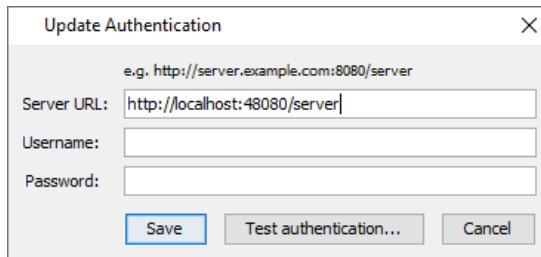


Figure 138. Update Authentication

5.3. File System Import

When creating a new *File System Import* Configuration, a unique *Name* must be defined. The *Directory to watch* can be a local or remote file location. The *File pattern* supports any TIFF image format and image-based PDF documents. For multiple file patterns, a pipe '|' separator must be used. The *File modification time threshold* is a time threshold value that prevents from importing files that are still open by the source application.

In the *Batch creation options*, a drop-down list with all published Jobs is available. the *Status* and *Priority* fields are optional Job properties. The *Max files per batch* option can be used to set a fixed number of files to be imported in every *Import Service* pull cycle. Furthermore, a combination of Batch and Document creation methods can define the actual Batch structure.

The *Batch creation method* options are:

- Create a single Batch, using all available files, or
- Create multiple Batches, one for each subdirectory

and the Document creation method options are:

- Create a single Document, using all available files, or
- Create a new Document, for each picked file, or
- Create multiple Documents, one for each subdirectory

Finally, the *Post processing configuration* rules apply to both successful processing and (permanent) failures. Again, when moving the imported files, the destination folders can be local or remote. All *Import*

Service start / stop actions and Batch creations are logged in `log\importd.log`.

Filesystem Import Configuration

Name:

Description:

Directory to watch:

Watch period: seconds

File pattern:

Match case

File modification time threshold: seconds

Only files that have not been modified within the given threshold period, will be picked

Batch creation options

Use Job:

Status:

Priority:

Batch creation method:

Document creation method:

Max files per batch:

Post processing

When processing is successful:

Move files to directory:

Purge after hour(s)

Delete files

When error happens:

Retry up to times, before marking as permanent failure

Keep history of errors for days

When permanent failure happens:

Move files to directory:

Purge after hour(s)

Delete files

If some files could not be post-processed (after either successful or unsuccessful import), they should not be picked for import in subsequent poll cycles. In this (rare) case, a file named **Ignored Files.txt**, within this watch directory, will list files or directories that should be ignored when checking for new files to import. The *Keep history of errors* parameter is the configured number of days that these records remain in the **Ignored Files** list. If a file has been ignored for this period, and still exists in the directory, it will be picked up again for import. It is therefore necessary to periodically check this file and remove old files from the directory.

5.4. XML File Import

When creating a new *XML File Import* Configuration, a unique *Name* must be defined. The *Directory to watch* can be a local or remote file location. The *XML File pattern* can be used to set specific file name patterns to be imported. For multiple file patterns, a pipe '|' separator must be used. The *File modification time threshold* is a time threshold value that prevents from importing files that are still open by the source application.

The provided XML files should meet the Info Input Solution XML Schema, more information about the XML Schema can be found in the [Info Input Solution XML Schema Reference.pdf](#)

Additional XML Schema references can be provided by using the Custom Batch Provider, adding a new Custom Batch provider is out of the scope of the Info Input Solution Administration Guide and it is described in detail in the [Info Input Solution Developers Guide](#).

Finally, the *Post processing configuration* rules apply to both successful processing and (permanent) failures. Again, when moving the imported files, the destination folders can be local or remote. All *Import Service* start / stop actions and Batch creations are logged in [log\importd.log](#).

XML File Import Configuration X

Name:

Description:

Directory to watch: Browse...
 Look in sub-directories

Watch period: seconds

XML file pattern:
 Match case

Custom Batch provider: Add/Edit Configuration

File modification time threshold: seconds
Only files not modified within the given last seconds will be picked.

Post processing

When processing is successful: Move files to directory: Browse...
 Purge after hour(s)
 Delete files

When error happens: Retry up to times, before marking as permanent failure
Keep history of errors for days

When permanent failure happens: Move files to directory: Browse...
 Purge after hour(s)
 Delete files

OK Cancel

5.5. Email Import

When using the *Email Import Configuration*, the following parameters should be set. A unique name should be used in the *Name* field. In the *Server URL* field a URL with the Email Server configuration should be used. The *URL Templates* can be used to easily set the Email Server information. In case an Exchange Server is used. The *Auto Discover URL* button can be used to automatically set the URL that is to be used. In this case the *Username* and the *Password* should be set before pressing the *Auto Discover URL* button. The *Watch Folder* is the folder that is to be watched for new items that are to be imported to

Info Input Solution. Please note all the items that will be found in the watch folder will be imported in each *Import Service* pull cycle.

Batch Creation options: the *Use Job* option can be used to set the job that the imported batches will be created with. The *Status* can be used to set a predefined status for the Imported batches. The *Priority* will set the priority of the imported batches. The following *Batch creation methods* are available.

- *Create a single batch using all emails in a single poll cycle.* This will add all the emails in one batch.
- *Create multiple batches, one for each email.* This will create multiple batches one for each email.

The *Post processing options* can be used to set the actions that will follow after a successful or an unsuccessful Email Import attempt. The available options are to delete the Emails or move them to another folder. The target folder should be available in the Email Account that is used.

Note, the emails that will be imported will follow the Job Import options, for more information please see from the [Job Import Options](#)[The options that will be followed when importing files](#) the section *Outlook Email Drag-n-Drop & Import Options*.

Email Import Configuration X

Name:

Description:

Server URL: URL templates Auto discover URL

Username:

Password:

Watch Folder:

Watch period: seconds Test connection...

Batch creation options

Use Job:

Status:

Priority:

Batch creation method:

Post processing

When processing is successful:

Move emails to folder:
 Purge after hour(s)

Delete emails

When error happens:

Retry up to times, before marking as permanent failure
Keep history of errors for days

When permanent failure happens:

Move emails to folder:
 Purge after hour(s)

Delete emails

OK Cancel

Figure 139. Email Import Configuration

5.6. Box Skill Import

Box Skill Import can be used to import files from Box. In order to perform this configuration it is required to have a Box Skill application configured at the Box side to monitor the required folders. Then the Import Service will deploy a Tomcat Server, it is required that the Tomcat is accessible from the Internet from HTTPS protocol in the default port 443.

5.6.1. Box Skill Configuration

The steps below can be followed to configure a Box Skill from the Box side. Note, the instructions below are subject to change at Box site's discretion.

1. First of all create a custom Box Skill, navigate to the Developer Console > Create New App > Custom Skill,
2. Then go to Admin Console > Apps > Custom Apps and click Authorize New App to authorize the new box skill,
3. Keep the Skill's Client ID. This can be acquired from developer console from the page where the skill was created,
4. Configure the skill from the Admin Console open the following URL,

<https://app.box.com/master/skills>

1. Add the folders you want to monitor with the skill setup

5.6.2. Box Skill Import Service Configuration

The following options needs to be configured,

- *Name*: a name for this import configuration
- *Description*: a description for this import configuration
- *File filtering*: Use this checkbox to filter out the files that comes from specific applications or users.
- *Skill Listening Port*: The port that will be used by the Tomcat that will deployed by the *Import Service*.
- *Download Directory*: A temporary directory that will be used until the files are imported to the system.
- *Add Box Import Metadata*: If the Box metadata will be added in the document as custom properties.

The *Batch creation method* options are the following:

- *Use Job*: The Job that will be used to create batches.
- *Status*: The status of the created batches.
- *Priority*: The priority of the created batches.

Finally, the *Post processing configuration* rules apply to both successful processing and (permanent) failures. When moving the imported files, the destination folders can be local or remote. All *Import Service* start / stop actions and Batch creations are logged in [log\importd.log](#).

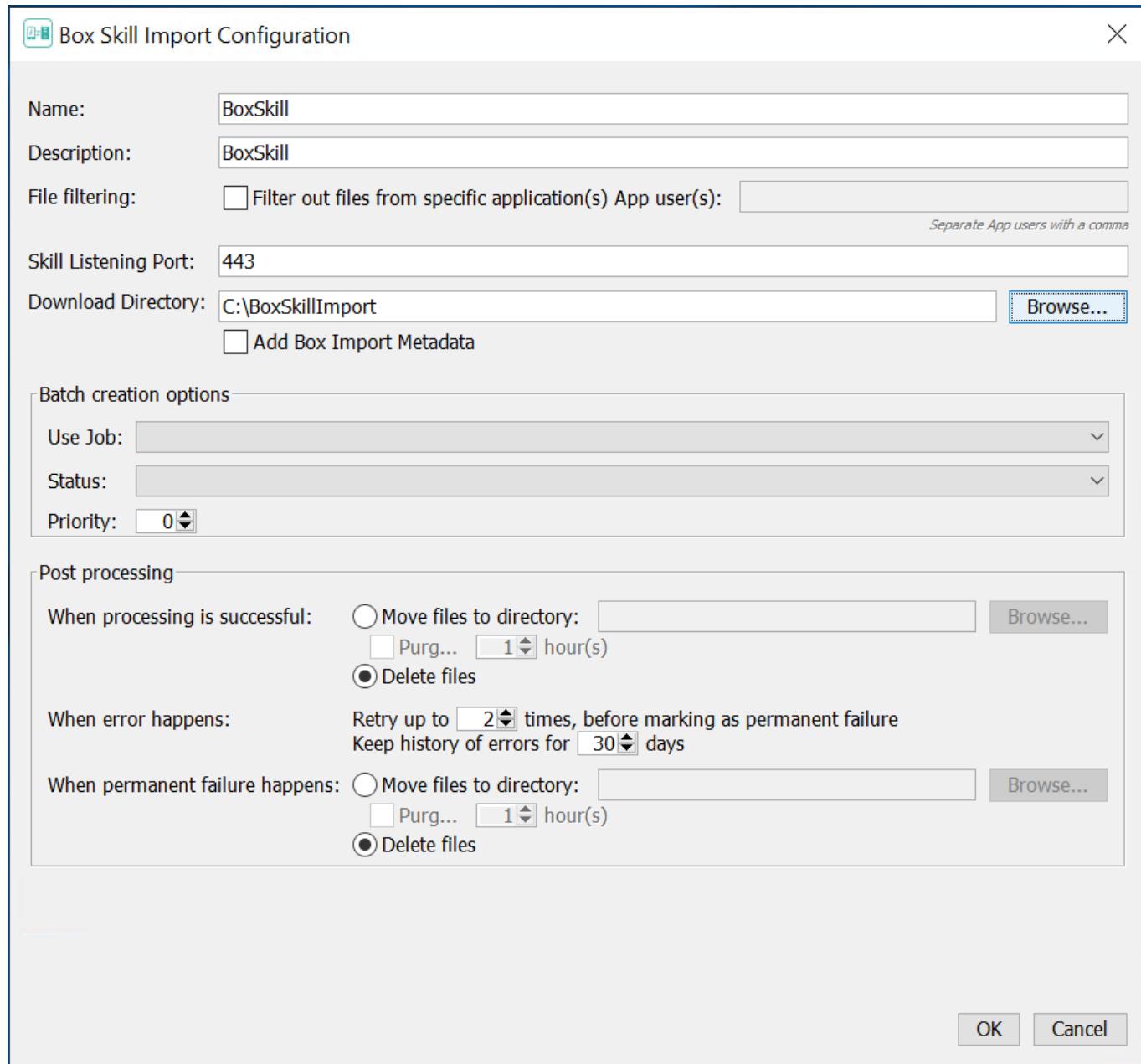


Figure 140. Box Skill Import Configuration

5.6.3. INfuse Import

The *INfuse Smart Connected Scanning Solution* offers network-connected capture that sits at the front edge of an organization's process and integrates directly into a line of business system. Content can be onboarded directly into business processes and real-time acknowledgement received at the point of

scanning. Thanks to immediate exception notifications, missing forms, signatures, and other crucial information can be caught and corrected instantly, saving money and improving the customer experience.

The Import Service can connect directory to the *INfuse Smart Connected Scanning Solution* and create batches in Info Input Solution by using the scanned images.

5.6.3.1. INfuse Import Configuration

The following items needs to be configured in the *Import Service INfuse Import* configuration.

- *Name*: Enter a Name for your configuration
- *Description*: Enter an Organization ID and a Plugin Key provided by Kodak Infuse based on your license, its similar having a username/password, but in this case those are used to get authenticated at Infuse.
- *Organization ID*: Pick a download directory for the files imported from Kodak scanners to be locally saved.
- *INfuse URL*: Define the Infuse URL and the Destination Service URL provided by Kodak.
- *Missed Documents Interval*: This is defining the period between executions of retrieving any missed documents from DS in seconds. Allowed range is 120-1200 seconds (2 to 20 minutes), with a default value of 120 seconds. This is configurable for performance reasons, in order not to let it consume a lot of missed documents at once, that may lead to peaks at Infuse and our side also.

Finally, the *Post processing configuration* rules apply to both successful processing and (permanent) failures. When moving the imported files, the destination folders can be local or remote. All *Import Service* start / stop actions and Batch creations are logged in [log\importd.log](#).

Infuse Import Configuration

image::branded_kiis/Import_Service_infuse_Import_GUI.png[pdfwidth=100%,align="center"]

6. Export Destinations Management

Export Destinations are modules that export *Batch* data to external systems. They are used by *Export Configurations* which customize their behavior.

Export Destinations can be managed from the *Export Destinations tab*, in the *Setup data dialog*:

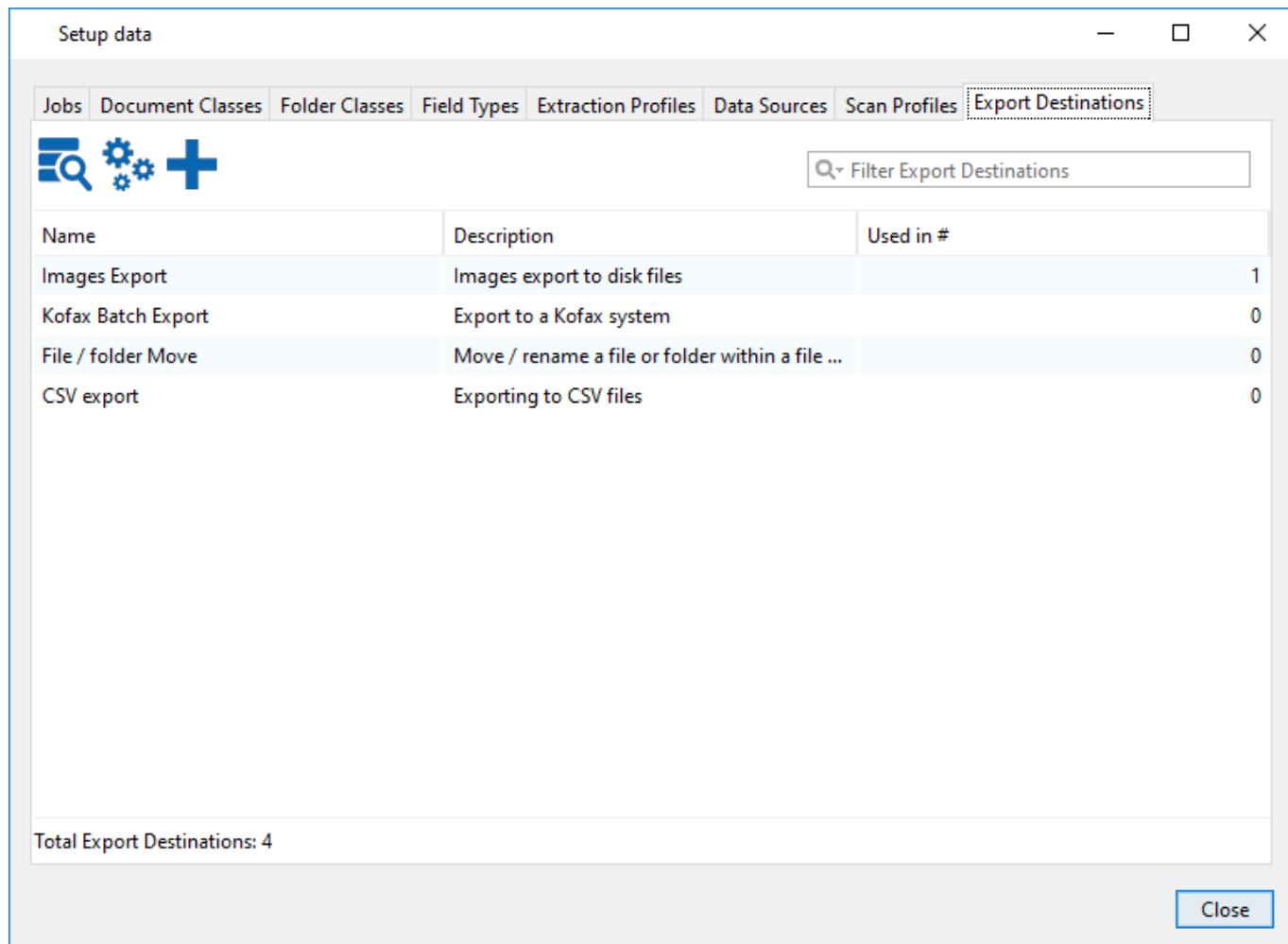


Figure 141. Setup data dialog: Export Destinations tab

Info Input Solution comes by default with three *Export Destinations* that are ready for use:

- [Images Export](#) allows you to export Images to a local directory.
- [Kofax Batch Export](#): allows you to send a *Batch* to Kofax, using the *Kofax Capture Import Connector*.
- [File / folder Move](#): allows you to move or rename a file or folder, within a file system.
- [CSV Export](#): allows you to export the *Batch* structure, in text or CSV format, to a local directory.

There are also two *Export Destination Templates* that must be setup before becoming available for use:

- [Database Export](#): allows you to export the *Batch* structure to a remote database.
- [Disk Export](#): allows you to export the *Batch* structure, in XML, to a local directory.

The *Export Destinations tab* displays only the *Export Destinations* that are setup and ready to be used. This is the reason why the *Export Destination Templates* are not shown on the *Export Destinations tab*.

6.1. Export Destination Templates

A *Export Destination Template* is used as the basis to create new *Export Destinations*. To create a new *Export Destination*, click on the *Create new Export Destination from Template* button. This will bring up a dialog where you must select which *Export Destination Template* to use:

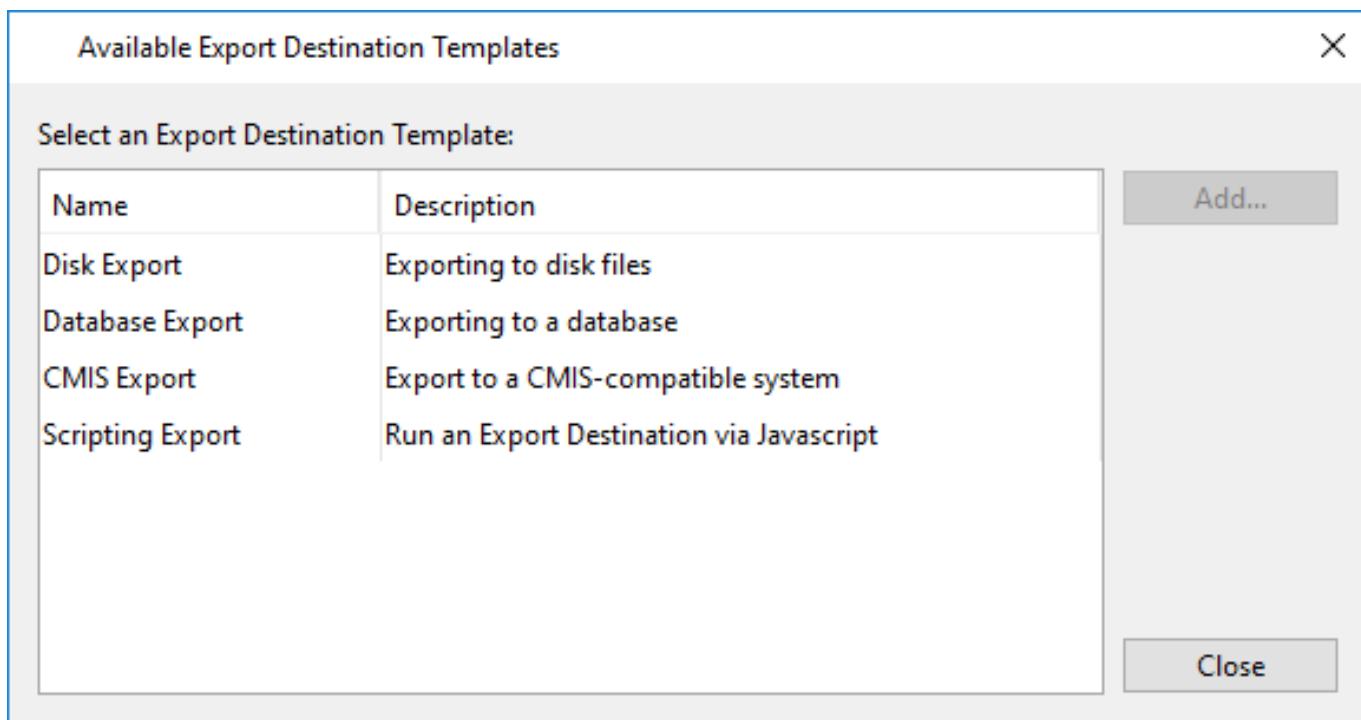


Figure 142. Export Destination Templates selection dialog

Select the *Export Destination Template* you want to use and press *Add...* This will bring up the corresponding dialog for creating an *Export Destination*:

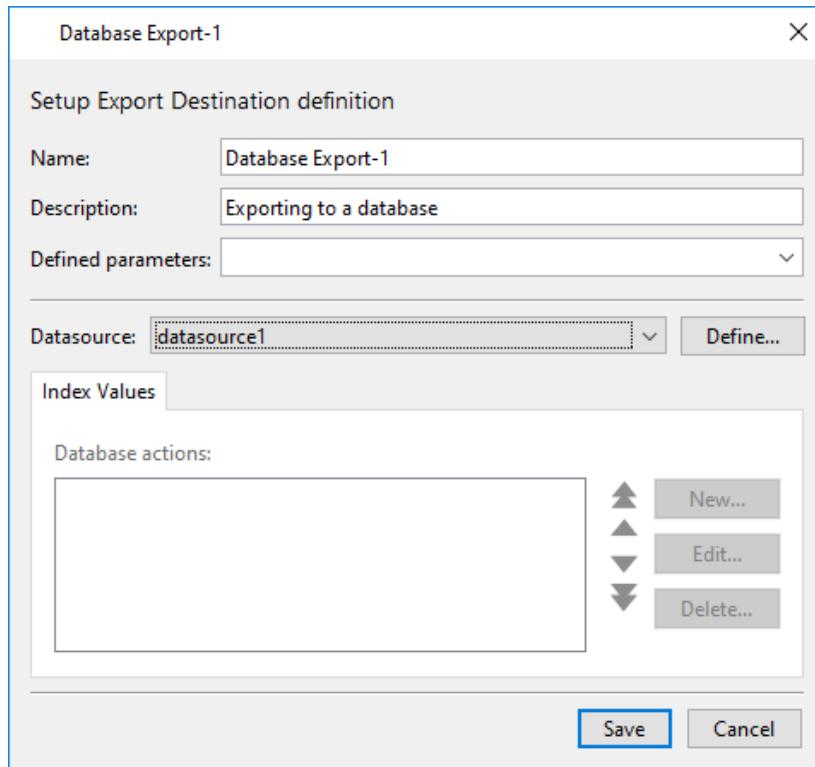


Figure 143. Export Destination Templates selection dialog

After defining all required parameters you can press *Save*. This will create a new *Export Destination* which will be shown on the *Export Destinations* tab.

You may use the same *Export Destination Template* to create multiple *Export Destinations*. For example, you may use the *Database Export* to create an *Export Destination* for a MSSQL database and another one for an Oracle database.

6.2. Editing Export Destinations

You may edit any *Export Destination* that appears on the *Export Destinations* tab. It is also possible to create copies of *Export Destinations* using the *Duplicate* popup menu item. For example, this may be useful if you need to create different flavors of the *Images Export*, one that produces TIFF files and one that produces PDF files.

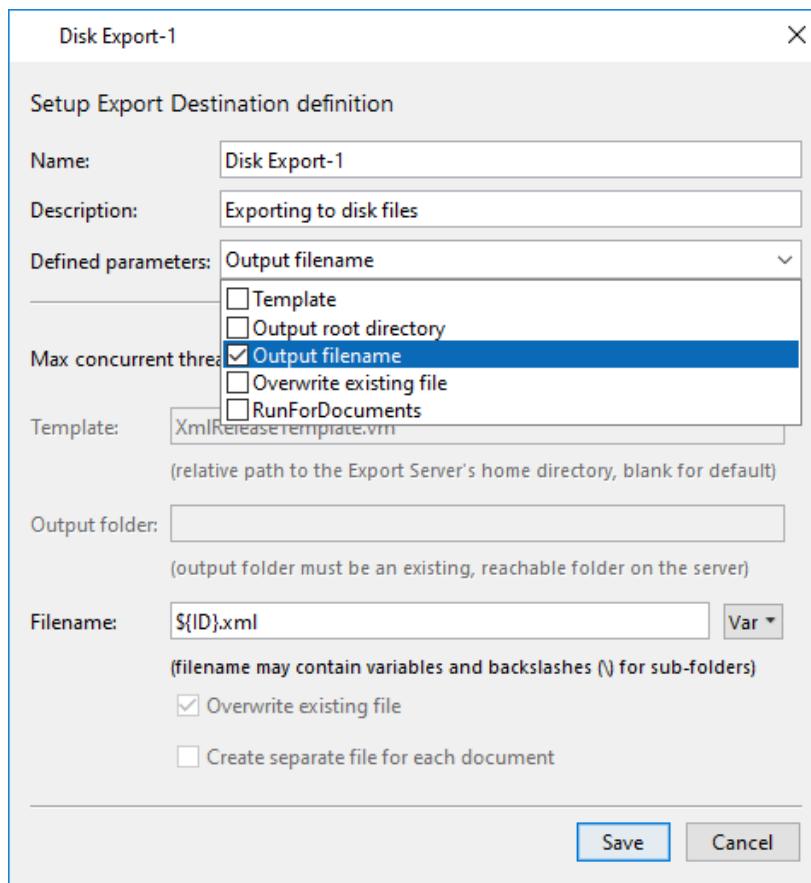
Each *Export Destination* contains a set of parameters that need to be setup. For example, the *Images Export* destination has the following parameters: *File format*, *Compression for binary images*, *Compression for color images*, *JPEG quality*, *Annotations*, *Output folder*, *Filename*, *Multipage*, etc.

So, the setup dialog for each type of *Export Destination* is slightly different. The top part of the dialog contains the fields *Name*, *Description* and *Defined parameters*. The *Defined parameters* combo box displays a list of all the parameters of the *Export Destination*.

The bottom part of the dialog contains all the parameters that can be configured.

Notice that the *Defined parameters* list contains all the available parameters. You need to check those parameters that you want to define (ie. hard-code) in the *Export Destination* you are editing and then provide the values below. The parameters that you check on the *Defined parameters* box will be enabled on the bottom part and you will need to fill-in their values.

For example, if you are creating an *Export Destination* that produces TIFF files, you may select to define the *File Format* and *Multipage* parameters. Note that all the remaining parameters are disabled on the bottom part of the dialog.



The parameters that were not selected on the *Defined parameters* box, must be defined by the *Export Configuration* that uses this *Export Destination*. In certain cases, an *Export Configuration* may override a parameter defined by the *Export Destination*.

Note that in the example above the *Output Folder* is not defined by the *Export Destination*. Any *Export Configuration* that uses 'Tiff Images Export' will need to provide its own value for the *Output Folder*. If on the other hand you decide to define the *Output Folder* on the *Export Destination*, then all *Export Configurations* that are based on it will inherit this value for the *Output Folder*.

The relationship between *Export Destinations* and *Export Configurations* allows you to easily change some basic parameters on the *Export Destination* and have this change affect all related *Export Configurations*. For example, in the case of *Database Export*, it makes sense to always setup the *Datasource* in the Export Destination then create custom configurations upon that. The advantage is that if you need to change the *Datasource* you only need to do it in a single place. So, if you plan to have 10 or 20 different targets on the same database, you should first create the base *Export Destination* where you defined the *Datasource* and then use that to setup your *Export Configurations*.

6.2.1. Output variables

Export Destinations may have *output variables* which are set at runtime and can be used as input variables to the next-in-sequence *Export Destinations*. For example, the *Images Export Destination* defines the **\$ImageOutputPath** variable after it is being executed: this variable is then available to use in the next *Export Destinations* (for example, a *Database Export destination* that is executed later may take this value and insert it in a database table).

6.3. Images Export

The *Images Export Destination* exports images to a local directory.

The following parameters are available:

Name

Name to be used to refer to this *Export Destination*.

Description

A brief description.

File format

The format of the image files that is required. Not all formats are available for all types of images. For example, *Black & White images* cannot be saved as JPEG files. The TIFF output format is usually the best choice here because it supports all types of images and it also supports *Multipage* files.

Compression for color images

Select the required compression for color images.

JPEG quality

If JPEG was chosen from the previous option, the quality level needs to be also selected. Valid values are from 1% to 100%. A value above 70% is recommended.

Annotations

Choose what should be done with *Annotations* that may exist on the images: either '*Burn*' them on the image, or *Ignore* them.

Create separate xml files with annotation details

Separate XML files containing information about the annotations on the images can be chosen to be created, whether they are superimposed (burned) onto the image itself or not. One XML file will be created for each image file.

Output folder

This is the base folder where the images will be stored: sub-folders may be created below that base folder, by using appropriate patterns in the *Filename* property.

Filename

The filename of the output image should incorporate various variables in the file name pattern. By clicking on the Var button on the right, a list of available [variables](#) can be viewed. Using backslashes (\) in the pattern facilitates the creation of sub-folders for the output images. It's important to note that the filename extension should not be included here as it will be automatically appended based on the output file format.

Export Level

Defines how the Images Export will be executed. The following options are available,

- *Export each document separately*: This is the default option, one multi page file will be created per *Document*.
- *Export each page separately*: This option will create one file per page.
- *Export the task as a whole*: This option will create a multi-page file per export level, for example if the *Images Export* is configured in the *Batch* it will create one multi-page file with the images from the whole *batch*. If the *Images Export* is configured in the *Folder* it will create one multi-page file per *Folder* that will contain all the images of each *Folder*.
- *File arrangement options*: Specify if you want to *skip first page of each document*.

If file already exists

Specifies the action that will be taken in case that a file with the same name is found in the *Output folder* that has been previously configured. The options available are:

- *Create a new file*: Creates a copy of the existing file with a unique name.
- *Send this task to error*: Task is sent to error mode and the batch stops being processed.
- *Skip this file*: Current file is skipped, the error is logged, and the remaining files of the batch are processed.
- *Overwrite this file*: The existing file gets overwritten with the current file.



You can use the same *Export Destination* more than once: so it is very easy to output images twice, one time without annotations and one time with annotations 'burned' on them. You just need to setup two *Image Export Destinations* and choose the appropriate options.

6.3.1. Output Parameters

The *Images Export Destination* has one output parameter called `$ImageOutputPath`: the output folder is saved in this parameter.

6.3.2. Filename pattern examples

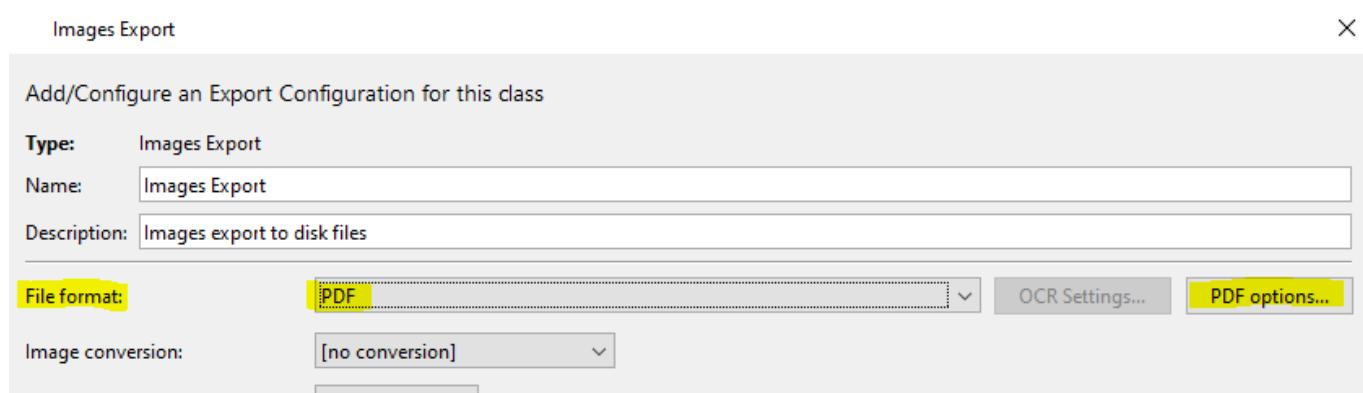
The *Filename* parameter supports expressions (patterns) so you should carefully choose the format in order to group your output in directories. Here are some examples of filename patterns and what the output images will look like:

Filename pattern	Explanation	Sample output image files
<code> \${ID}</code>	Uses the internal ID of each node as the filename	<code>\1.tiff \2.tiff \3.tiff</code>
<code> \${ParentID}\\${ID}</code>	Uses the parent ID to create a subfolder first	<code>\1\2.tiff \1\3.tiff \1\4.tiff \5\6.tiff</code>
<code>Batch-\\${BatchID}\Doc-\\${AbsoluteIndex}</code>	Uses the Batch ID to create a sub-folder, then puts all the documents (multipage) in a single folder	<code>\Batch-1\Doc-1.tiff \Batch-1\Doc-2.tiff \Batch-1\Doc-3.tiff</code>
<code>Batch\\${BatchID}\Folder\\${FolderAbsoluteIndex}\Doc\\${DocumentAbsoluteIndex}\Page\\${AbsoluteIndex}</code>	This example is used for Jobs with 4 levels and with no-multipage output. There is one image exported for each page. (BatchID=2718)	<code>\Batch2718\Folder1\Doc1\Page1.tiff \Batch2718\Folder1\Doc1\Page2.tiff \Batch2718\Folder1\Doc2\Page3.tiff \Batch2718\Folder1\Doc2\Page4.tiff \Batch2718\Folder2\Doc3\Page5.tiff</code>

6.3.3. PDF with Text layer and Metadata

If one of the PDF File formats (PDF, PDF/A, PDF+Text) is selected in Images Export configuration, a new

set of options is unlocked at the Images Export window, as shown below:



These extra options include the following sections:

PDF text layer

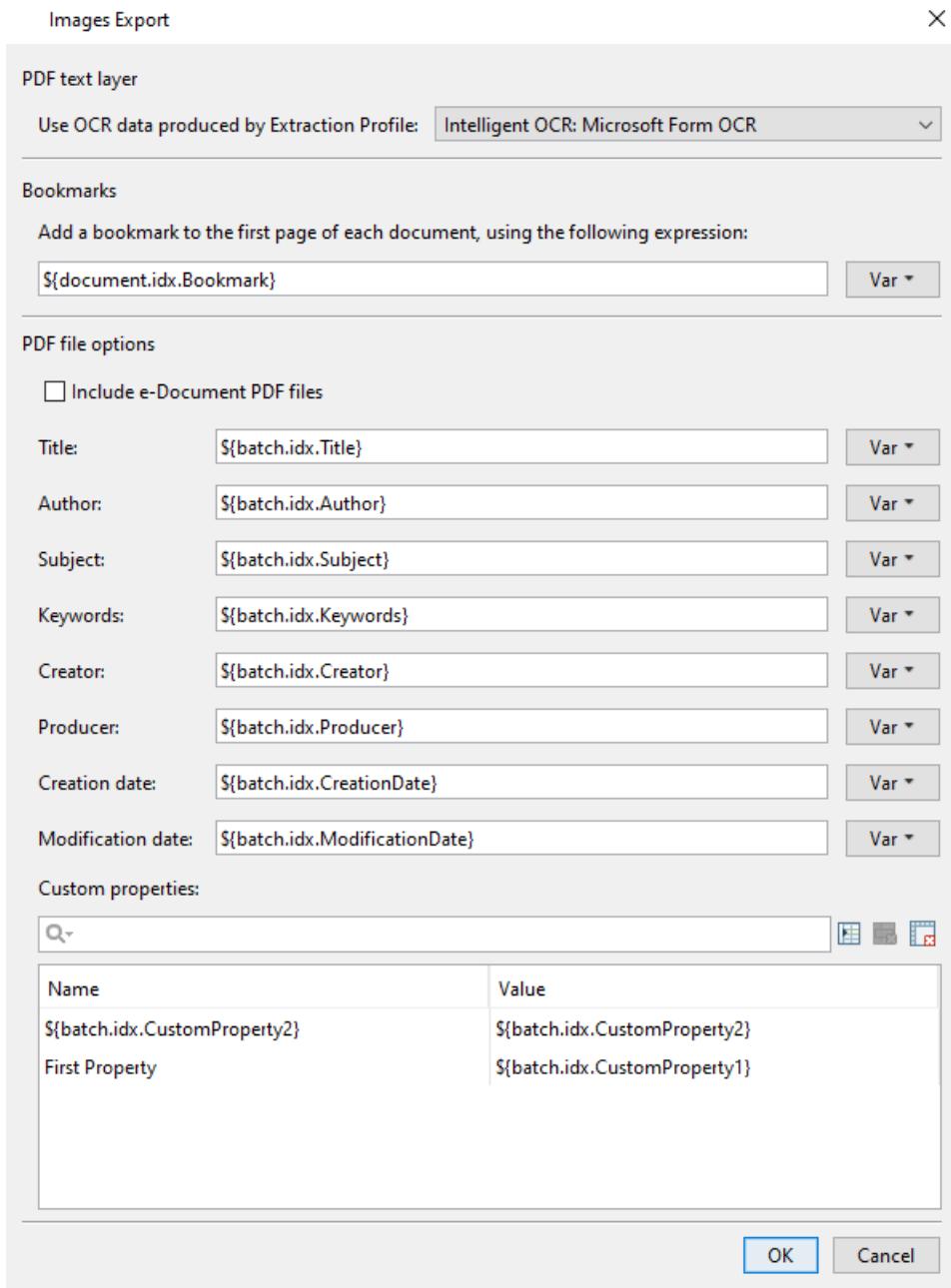
This option can be used to generate PDFs with text layer (searchable) using OCR data from a previous Intelligent OCR Workflow Step. A full example can be found at [Creating an Intelligent OCR Project](#) section.

Bookmarks

This option allows the user to create PDFs with bookmarks by setting an expression as the bookmark name. The expression should be a standard field, property or system variable (or a combination of those).

PDF file options and properties

Custom PDF properties added to the created PDF, visible in PDF viewers.



6.4. Kofax Export

The *Kofax Export Destination* is used to output all meta-data of a *Batch* and the associated images to a *Kofax Capture System* using the *Kofax Import Connector Server* add-on of Kofax. This destination will only produce valid results if setup for the *Batch* level.

The following parameters are defined:

Kofax Web Service URL

The Kofax Web Services URL (something like <http://<host>/AscentCollectionServer/api.asmx>)

Username of Kofax User

The **username** to connect to *Kofax Import Connector server*

Password of Kofax User

The **password** to connect to *Kofax Import Connector server*

Override with Kofax imported settings

Insert the value **true** here only for Jobs that are in *Kofax Linked* status, e.g. their name should have the *Kofax Linked* label at the end. You should use this only when you want to export a Batch to the same server you used to import the Job definition (see [Import/Export Jobs](#) for more info). Notice that a default *Kofax Export Destination* is automatically created for the batch level whenever you import a batch from Kofax.

Exporting to a *Kofax Capture System* supposes that the names of associated *Document Classes*, *Folder Classes*, *Field Types*, etc are compatible between the two systems. In practice, the only way to achieve the full uniformity of all the names of all the entities that are mapped between the two systems, is to use this destination in combination with a batch that has been imported from Kofax and setting the *Override with Kofax imported settings* property to **true**.

6.4.1. Output Parameters

There are no output parameters for the *Kofax Export destination*.

6.5. File / folder Move

The *File / folder Move Export Destination* is used to move or rename a file or folder within a file system.

The following parameters are defined:

File or folder to move/ rename

The name of the source file or folder.

Move to or new name

The destination file or folder name.

6.5.1. Output Parameters

There are no output parameters for the *File / folder Move Export Destination*.

6.6. Disk Export

The *Disk Export Destination* is used to output all the metadata of a node (*Batch*, *Folder*, *Document*) in a single XML file.

There is no *Disk Export Destination* initially, only a *Disk Export Destination Template*. When you create a new *Disk Export Destination*, you must give a new name and select the *Max concurrent threads* parameter. It would be good to also check the *Defined parameters* → *Template* option in order to hard-code the *Template* in the definition here.

The only parameters you need to define are the *Output folder* and the *Filename*. Please see the [Images Export](#) for a discussion on these two variables and their possible values.

The *Template* property refers to a `.vm` file that exists on the server directory `<Info Input Solution Install Root>/fd/`. This is actually a file that contains a [Velocity Script](#). Velocity is a scripting engine from Apache which is used by the *Export Service* to output the XML file. You can read the [Velocity User's Guide online here](#).

If you want to modify the XML that is produced by the *Disk Export Destination*, you can create a new velocity script by copying the `<Info Input Solution Install Root>/released/XmlReleaseTemplate.vm` file and change its contents accordingly to match the desired output. The Velocity scripting language is versatile and allows you to modify the output in many different ways.

6.6.1. Output Parameters

There are no output parameters for the *Disk Export Destination*.

6.7. Database Export

The *Database Export Destination* is used to write/insert the meta-data of a node (*Batch*, *Folder*, *Document*) to a remote database. It cannot be used to insert images to the database.

There is no *Database Export Destination* initially, only a *Database Export Destination Template*. When you create a new *Database Export Destination* you must give a new name and select the *Datasource* parameter.

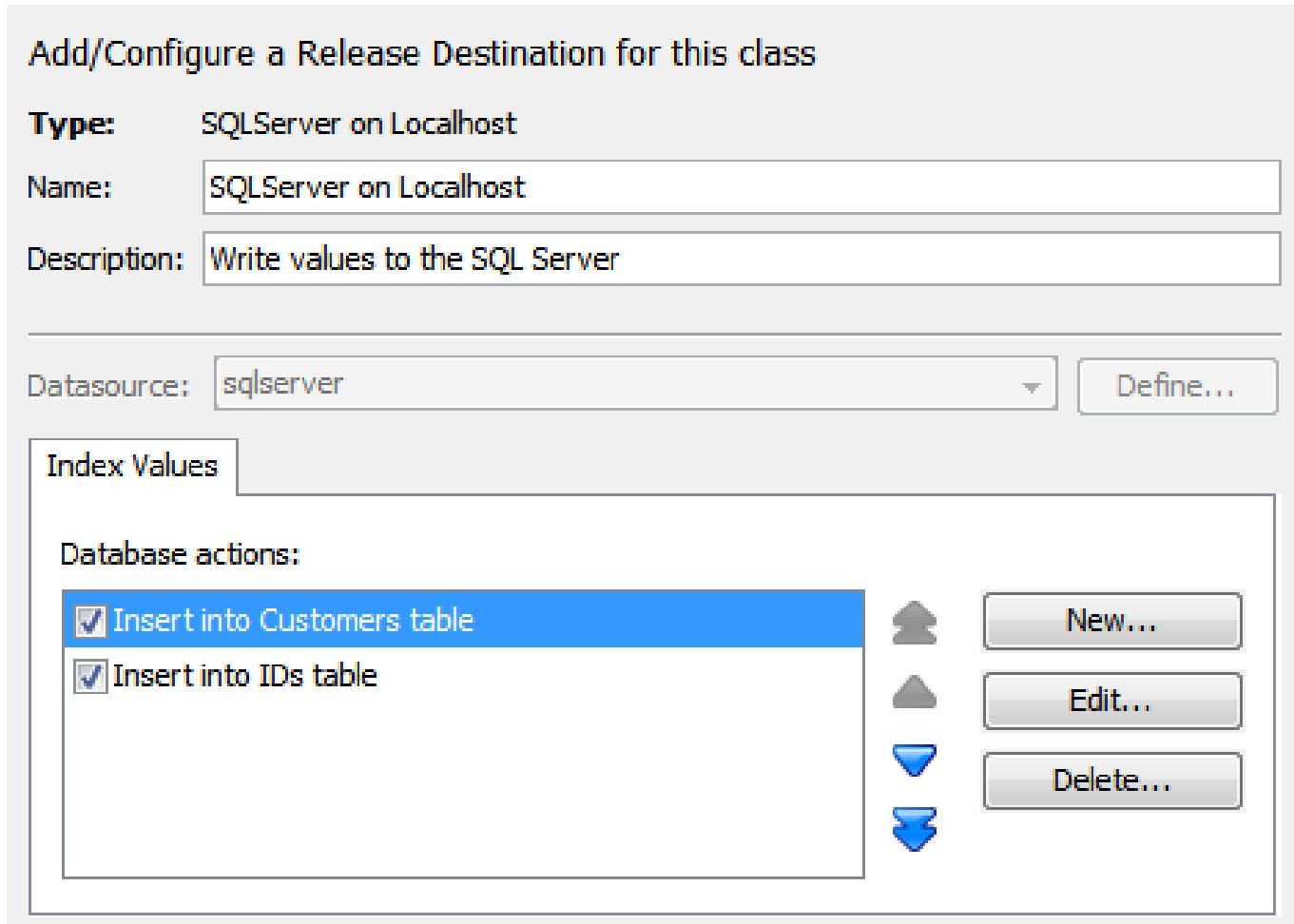


Figure 144. Database Export setup dialog

The *Database Export destination* performs one or more *Database actions* to a remote database: each action corresponds to a SQL INSERT statement that is build dynamically. Click on the *New...* button to add a new *Database action*.

From the *Database Action definition dialog*, enter a *Name* for the action and choose the table you want to insert data into. The *Column mapping table* below is populated once you have selected the *Table* parameter. The left column of the table has a list of all available columns of the selected table. The right-column is where you can choose which value you want to map/insert to each column.

Clicking on a cell in the right-column gives you a drop-down list that allows you to select from possible values. The list contains all the *Index fields values* for the level you are adding this destination for, standard *Node values* (see [Variables](#)) and *Output variables of previous export configurations*:

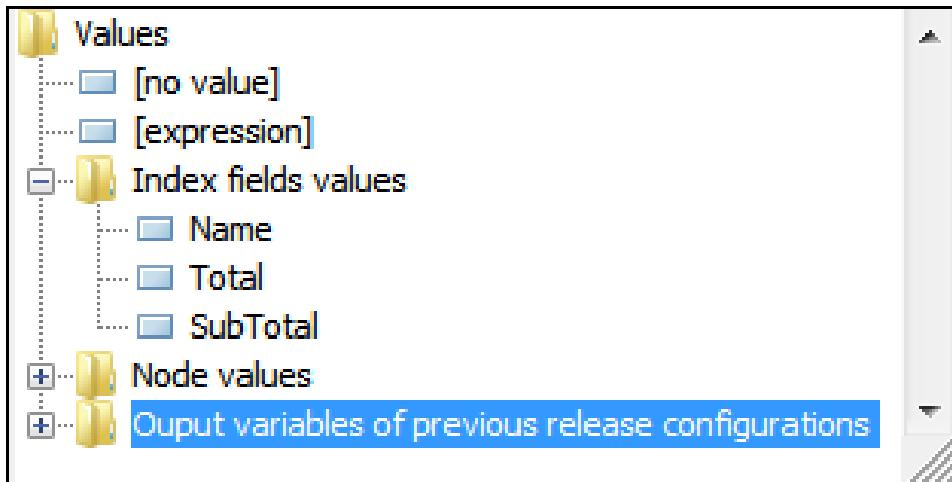


Figure 145. Database Action definition mapping list: Values list

Selecting the **[expression]** from the list above, gives you a dialog where you can manually enter a *custom expression* that contains more than one values and/or constants. For example, if you want to map to a single column of a table both the **Total** and **SubTotal** index fields separated with a dash (-), you can write the expression **\$Total-\$SubTotal**.

Select the **[no value]** to clear the mapping for a column.

6.7.1. Output Parameters

The *Database Export Destination* has one output parameter called **\$InsertId**, which holds the database identifier (number) of the last insert operation.

6.8. ECM Export

The ECM Export (together with the Images Export) extends the value of scanned images by facilitating their reuse across delivery channels. The ECM Export enables Info Input Solution to export all newly captured images and meta-data directly into a specified Enterprise Content Management environment. These images can include graphic formats (e.g., JPG, GIF) as well as text items generated through OCR during capture. Once images are in an ECM Repository, they can be modified, started on business workflows, integrated with and linked to other content in the system.

The ECM Export and Images Export work together. The Images Export defines the location of the images to be exported to the ECM Repository by the ECM Export. Therefore, before defining the ECM Export, the Images Export has to already be defined and added under the Export Destinations for the folder or document.

Setup a new Release Destination definition

Type:	ECM Release
Name:	ECM Release
Description:	Release to ECM Repository
Defined parameters:	▼
Webservices URL	
ECM Server	ICMNLSDB
ECM User	ICMADMIN
ECM Password	*****
ECM Type	ICM
Images Directory	C:\Aisha\ECM

Figure 146. Pop-up menu options in batch manager

Figure : ECM Export Configuration Panel

1. To define an ECM Export; Launch the export destinations window by clicking "Export" on the Folder or document level. (Batch level is not supported);
2. Click "Add/List";
3. Select ECM export from the list of export destinations;
4. Click "new" to define a new images export configuration;
5. On the ECM Export panel;
 - a. Name: Enter a name to be used to refer to the images export. The name has to be unique for each export defined on the system.
 - b. Description: Enter a brief description of the export.
 - c. Defined parameters: Select one or more options required. (Please note: depending on the options selected here, only the required fields in the next section will be enabled for the selected options.)
 - d. Webservices URL: Input the URL to the ECM web services.
 - e. ECM Server: Provide the name of the ECM Server (/or Repository)
 - f. ECM User: Provide the username of the user with privileges to create documents in ECM Repository.
 - g. ECM Password: Provide the password for the user account provided.
 - h. ECM Type: Enter the key (as shown in table below) for the ECM Repository to connect

ECM Repository	Key
IBM Content Manager	ICM
	Alfresco Community Edition
ALFRESCO (not yet supported)	
Alfresco Community Edition	ALFRESCO (not yet supported)

1. Images Directory: The value entered here has to be exactly the same as what was defined in the images export output folder.
2. Click OK.
3. Select the new ECM export destination and click "Add".
4. The new ECM export destination is added to the export destination panel.

Assumptions:

- Images Export has already been defined for the item at folder or document level.
- ECM Export has to be defined at the same folder or document level as the Images Export.
- The Images Export filename attribute is configured as follows;
 - For Images export on folder: filename is {FOLDERID}_{DOCUMENTID}
 - For Images export on document: filename is {DOCUMENTID}

6.9. Scripting Export

The *Scripting Export Destination* is an Export Destination that delegates all the release destination methods to javascript .

All release destination methods that are normally implemented in java (configure, start, stop, getMaxConcurrentThreads, release), need to be implemented with in javascript through destination/configuration parameters.

The main purpose of the Scripting Export Destination is to:

- Send emails before / after / between Export steps
- Execute any system scripts (.bat, .exe, .vbs etc) that do some certain customized work
- Change some batch property between Export steps
- Modify the Images Export output files immediately after they are created (File / Folder move does not cover all use cases)

6.9.1. Parameters

The following parameters are available:

Configuration Script: Contains the configuration methods configure, start, stop, getMaxConcurrentThreads

Export Script: Contains the runtime method release

6.9.2. Output Parameters

By default the *Scripting Export* will not output any parameters. However, it is possible to output a customizable export parameter by using the script below,

This example will create the custom variable ExportedItemID which will be available in the next Export Configuration. Note, the ExportedItemID is a variable that holds a string and needs to be initialized in the scripting export.

```
context.setSharedObject("ExportedItemID", ExportedItemID);
```

6.9.3. Example

Using a java process to execute a .bat file.

Configuration Script

```
importPackage(java.io);
importPackage(java.lang);
importPackage(Packages.scncommon.model.release);

// Global variables section
var emailExecutablePath; // The path to the .bat executable
var emailFrom; // The from mail address used by the .bat (vbs) script
var cmdExecutablePath; // The from mail address used by the .bat (vbs) script

function configure(configuration, log) {
    emailExecutablePath = "C:\\\\Temp\\\\BatchFiles\\\\ExecEMAIL.bat";
    log.info("Configured release with param emailExecutablePath =" + emailExecutablePath);

    emailFrom = '"info@imageaccesscorp.com"';
```

```
log.info("Configured release with param emailFrom = " + emailFrom);

cmdExecutablePath = "C:\\Windows\\System32";
log.info("Configured release with param cmdExecutablePath = " + cmdExecutablePath);
}

function getMaxConcurrentThreads() {
    log.info("Returning Max Concurrent threads no#");
    return 1;
}

function start() {

}

function stop() {

}
```

Export Script

```
function release(context) {
    log.info("Executing release with params " + emailExecutablePath);
    log.info("Executing release with params " + emailFrom);
    var document = context.getReleaseItem();

    if (document instanceof com.imagetrust.tc.model.object.IDocument) {
        var emailToIndexField = document.getField("Email").getValue();
        log.info("Executing release with index param " + emailToIndexField);

        var envp = ["resultlog=" + System.getenv("resultlog"),
                   "common=" + System.getenv("common"),
                   "Path=" + cmdExecutablePath];

        log.info("Executing release with param env param[0] = " + envp[0]);
        log.info("Executing release with param env param[1] = " + envp[1]);
        log.info("Executing release with param env param[2] = " + envp[2]);
    }
}
```

```
log.info("Processing document: " + document.getId());
var docFilePath = context.getSharedObject(ImagesReleaseCommon.OUTPARAM_PATH);
log.info("Executing release with Attachment File path: " + docFilePath);
var command = "cmd /c " + emailExecutablePath + " " + emailToIndexField + " " +
emailFrom + " \"Email from IT: Document: " + document.getId() + "\" " + "\"\" " +
docFilePath;
log.info(command);
var process = Runtime.getRuntime().exec(command, envp);
process.waitFor();
} else {
    log.warn("Expected document. Do nothing!");
}
}
```

6.10. CSV Export

The CSV Export Destination is used to output all the metadata of a node (Batch, Folder, Document) in a single CSV file.

There is a description below for each available field:

Name: A name to be used to refer to this *Export Destination*.

Description: A brief description

Defined Parameters: This field makes available each one of the fields below to be hardcoded, for every CSV Export Configuration.

Output Folder: A folder path that will be used to store the output file .

Filename: A desirable file name var's can be used

Create separate files for each document: By checking this checkbox a different csv file is created for each document.

If file exists: This drop down list offers three options

- *Fail:* cancels the export step and returns an error,
- *Overwrite:* overwrites the existing csv file,
- *Append:* adds the new information to the same file.

Output a line for each: do a line feed in any new Batch, Folder, Document, Page. Add line title adds an extra line with the corresponding title.

Delimiter: the character to be added for separating the csv's fields.

Text qualifier: a drop down list with three options ", ' , and none if the text inside will be between two apostrophes or quotation marks or nothing.

Output Fields: The fields that contain the metadata and will be exported to the csv. As shown in the picture below the the fields on the left are the available fields. The fields on the right are the fields that are selected

6.10.1. Output Parameters

There are no output parameters for the *Disk Export Destination*.

7. Batch and Workflow Management

The Info Input Solution Batch Manager component is used to review and manage the work available in the system.

7.1. Batch Manager

Batch Manager provides a comprehensive view of all batches in the system in a windows explorer-like environment. Use *Batch Manager* to:

- View the batches in the system
- Add or edit *notes* for a batch
- Change the *status* and *priority* of a batch
- Move batches to another [Queue](#)
- *Delete* batches
- View information about the number of folder/documents/pages in a batch

To access the *Batch Manager*, click on the *Tools & Options menu* → *Batch Manager...* item.

The *Batch Manager* provides a view of all the batches in the system. Access to the *Batch Manager* is granted to users with the [Batch Manager permission](#). *Private batches* are only visible in the *Batch Manager* only if the user also has the [Private batches of other users](#) permission.

The list of batches in the *Batch Manager* is automatically refreshed every 60 seconds: you may change the refresh rate, by right-clicking on the *checkbox* on the bottom-right of the window and entering a new value in the *Input dialog* that will appear. You can turn-off completely *auto-refresh* by un-checking the *checkbox* there. You may manually refresh the list of batches at any time by clicking on the *refresh* button on the toolbar.

The buttons on the toolbar perform these functions:

Table 1. Table : Batch Manager Toolbar

Button	Function
	<i>Refresh</i> : reloads the list of batches
	<i>Search</i> : displays the search bar at the bottom
	<i>Configuration</i> : displays a drop-down menu with several options to customize the list of batches

Button	Function
	Print: allows you to print the list of batches
	Exit: closes the Batch Manager window

7.2. Performing Actions on Batches

The *Batch manager* supports dynamic grouping of batches: you can drag any column to the area below the toolbar to group batches on that column. The figure below displays batches that are grouped first on the *Job*, then on the *Queue*. You can also sort batches by clicking on the column you want to sort on.

To manage a specific batch right click on the batch and select the task to perform:

Batch Manager																			
Step	Batch ID	Job ID	Batch des...	Job name	Folders	Documents	Pages	Created by	Group	Created	!	Mode	Status	Locked by	Locked on	Split from ...	Copied fr...	Private	New Batch
Step: Index (1 items)	27	5	2/4/1...	NewBatch...	1	1	1	admin	Administrat...	Feb 4, 201...	5	READY			0	4	No	No	
Step: Scan (1 items)	32	5	2/4/1...	NewBatch...	1	1	1	admin	Administrat...	Feb 4, 201...	5	READY			0	4	No	No	
Total batches: 2																			
<input checked="" type="checkbox"/> refresh in 49 s																			

Figure 147. Pop-up menu options in batch manager

Alternatively, you can select multiple batches or group rows (the ones in bold) if you want to perform an action on multiple batches. For example, in the above image, if you right-click on the *Job: Airbills (2 items)* line, then you will be performing the action on the 2 batches that belong to this group.

The *right-click menu* provides access to the following administrative functions:

Delete

Deletes the batch(es) from the system: this is a permanent, non-reversible action

Create Batch copy

Creates a copy of an existing Batch. This operation can be executed on multiple Batches which

belong to different Jobs or different published instances of the same Job. The selected Batches can also be in different steps in their respective workflows or in different modes. This operation cannot be executed on Batches which are in PROCESSING mode. If you need to create a copy from a Batch in PROCESSING mode, you need to first unlock it. If the Batch Manager User does not have the permission to create Batches, then he/she will not be able to create copies from existing Batches.

When the Create Batch copy is clicked, the 'Create new Batch from existing Batch' dialog is shown:

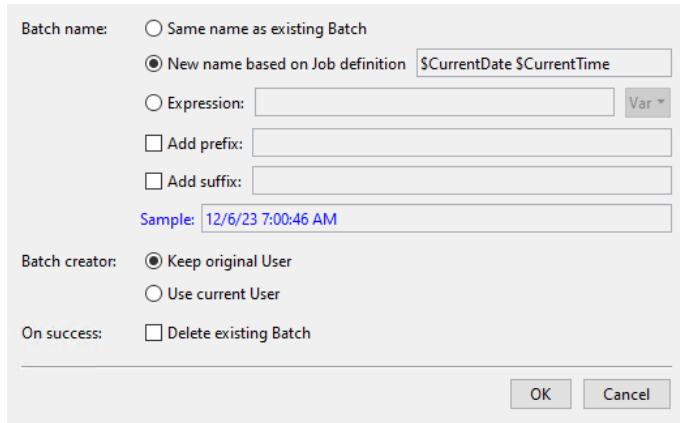


Figure 148. Batch Information Dialog

In this dialog you can configure how the Batch copies will be created:

Job

The Batch copies will be created using the Job selected in the combo box. The combo box will contain all published Jobs for which the Batch Manager User is allowed to create new Batches. Only the latest published instance of a certain Job can be used when creating Batch copies. Note that all Batch copies will use the selected Job and not the Job that the existing Batches belong to.

Workflow step

The Batch copies will be placed in the queue which corresponds to the selected workflow step. The combo box contains all workflow steps of the Job selected above. If you select one of these steps, then all Batch copies will be placed in the same queue and their mode will be READY. You can also select the 'Use current workflow step from existing Batch' option. In this case, an attempt will be made to match the name of the current step of the existing Batch, with the name of a step in the workflow of the selected Job. If a match is not possible, then the starting workflow step of the selected Job will be used and the newly created Batch copy will be placed in the starting step queue.

Batch name

Several options are available for the name that the Batch copies will take.

1. Same name as existing Batch: the name of the existing Batch will be given to the Batch copies.

2. New name based on Job definition: the 'Batch naming' expression of the selected Job will be evaluated for each Batch copy and the produced result will be the name of the Batch copy. As a reminder, the 'Batch naming' expression of the selected Job is displayed in the adjacent text box,
3. Expression: enter any custom expression which will be evaluated for each Batch copy. The produced result will be the name of the Batch copy. Note that the expression is evaluated on the Batch copy and not on the existing Batch. For example the expression `${JobName}` will return the name of the selected Job and not the name of the Job of the existing Batch.

In addition to the above three options, you may select to add a fixed prefix and/or suffix to the name of the Batch copy. Note that the prefix and suffix are fixed strings and not expressions that are dynamically evaluated for each Batch copy.

Batch creator

You can choose whether the User that will be assigned as the creator of the Batch copy, will be the same as the creator of the existing Batch, or the Batch Manager User who is executing this operation. Note that if an existing Batch is private, the Batch copy must also remain private and assigned to the same creator. So for private Batches you must use the option to keep the original User.

On success

You can choose to delete the existing Batch after the Batch copy has been successfully created.

Making a copy of an existing Batch is a complex operation because the Job of the original Batch may be different than the selected Job used to create the Batch copy. Incompatibilities between the two Jobs may prevent the Batch copy from being created. Please keep in mind the following when creating Batch copies:

- The Batch Manager's snapshot of an existing Batch must be up-to-date when the Create Batch copy operation is executed. In other words, if the Batch gets locked by a User or is changed/deleted between the time that the Batch Manager's data was last refreshed and the time that the operation is executed, then the Batch cannot be copied.
- If an existing Batch is private, but the selected Job does not allow private Batches, then the Batch cannot be copied.
- If an existing Batch is not private, but the selected Job allows only private Batches, then the created Batch copy will be private. If you choose not to keep the original User as the Batch creator, then it is possible that the Batch copy will not be accessible by its original creator.
- If an existing Batch contains e-Documents, but the selected Job does not allow e-Documents, then the Batch cannot be copied.
- The creation date for the Batch copy is the date (day and time) at the moment that the Batch copy is created and not the creation date of the existing Batch.
- The Batch status is copied from the existing Batch to the Batch copy, only if it is included in the 'Batch statuses' property of the selected Job.

- An effort is made to maintain the structure of the existing Batch. The result depends on the Batch levels (either Batch, Document, Page or Batch, Folder, Document, Page) of the Job of the existing Batch and of the selected Job. If both Jobs use the same levels, then all nodes are copied. If the existing Batch has Folders, but the selected Job does not allow them, then all Folders are removed and all Documents are placed under the Batch copy. If the existing Batch does not have Folders, but the selected Job uses the Folder level, then a single Folder is created and all Documents are placed under it.
- Page separation rules are ignored when creating Batch copies. For example, if the selected Job has a separation rule that creates a new Document every 5 Pages, the Batch copy will not be restructured in order to comply with this rule.
- All node properties are copied from the existing Batch to the Batch copy.
- All Page annotations are copied from the existing Batch to the Batch copy, only if the selected Job allows annotations. If the selected Job does not allow annotations, then all Pages in the Batch copy will have no annotations.
- No image modifications take place when creating Batch copies. For example, if the selected Job is setup so that its Batches have only black & white images, but the existing Batch has color images, then the Batch copy will also have color images.
- If the existing Batch contains a PDF file as an e-Document, but the selected Job treats PDF files as images, the PDF file will not be converted into images in the Batch copy.
- No scripts are executed when creating Batch copies.
- An effort is made to copy indexing information from the existing Batch to the Batch copy. If an Index Class (Document or Folder Class) is assigned on an existing Node, the system tries to find a matching Index Class, using the Index Class names. If a match is found, it is assigned to the corresponding Node of the Batch copy.
- No index field conversions take place when creating Batch copies. For example if the existing Batch had an Index Field which is String, but in the selected Job the same Index Field is a Date, the index field value will not be converted. This may lead to errors when the Batch copy is opened for indexing.
 - *Properties:* displays the batch's properties. Double clicking on a batch also displays the same *Batch information dialog*:

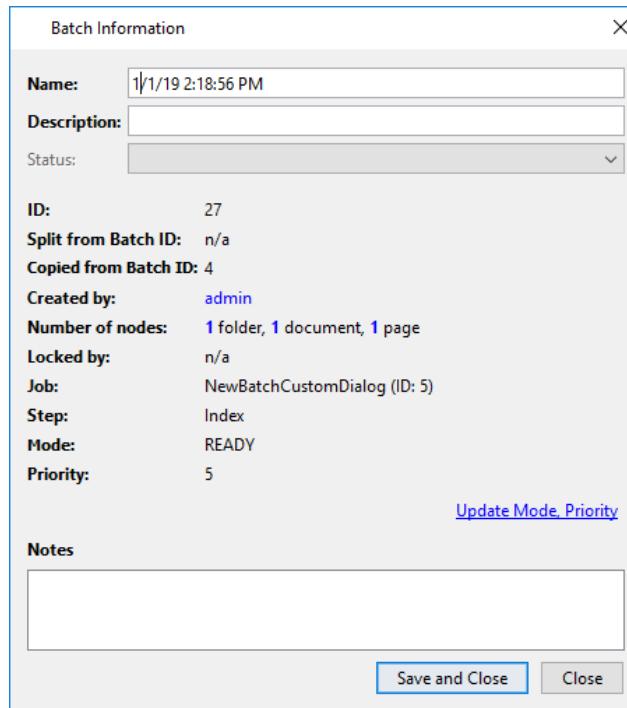


Figure 149. Batch Information Dialog

From this dialog, you can change the *name*, *description* and *status* of a *batch*, and also *add notes*.

- ° *Unlock*: unlocks the batch(es) and makes it available for processing. Whenever a batch is opened by a user, it is automatically locked. When a batch is closed or suspended it is automatically unlocked. Unlocking a batch using this function forces a batch to be unlocked thus forcibly taking the ownership from the current user: if you do that, the current user will be unable to close or suspend it and will need to just remove it from the workstation (the *Core Service* will not accept any changes the user makes to this batch). This action is useful in the case a *Client* crashes and the batch stays in the locked state for an indefinite amount of time.
- ° *Update Queue, Mode, and Priority*: updates the attributes of a batch. A batch can be either in the *Ready Mode*, meaning it is available to be processed, or in the *Error Mode*, meaning that some error has occurred during the last processing step and some manual intervention is required. The *Export Service* may set a batch in *Error Mode* if it failed to export it.
- ° *Unset private status*: removes the [private status of the batch](#).

When you select multiple batches to perform actions on, the *Core Service* performs these actions on separate transactions. So it is possible that some will fail and some will succeed. Upon completion, if some actions failed on some batches, a report with failed transactions will be generated and presented.

7.3. Customizing the Batch Manager

The customization options below are only available for the HTML Client

7.3.1. Pre-defined Batch Manager grouping:

The first available customization requires configuring the corresponding Server parameter by using comma-separated values of the names of Batch Manager columns.

1. Launch *Thick Client* and open the Server Configuration dialog from the *Tools & Options* Menu.
2. Search for the parameter `scanapp.client.html.batch_manager_grouped_columns`.
3. Create a comma-separated list using the names of the Batch Manager columns to group by the available column names are the following: `nodetype`, `nodeid`, `nodeorder`, `parentid`, `batchid`, `batchname`, `batchdescription`, `jobid`, `jobname`, `foldercount`, `documentcount`, `pagecount`, `creator`, `creatorgroup`, `createddate`, `priority`, `queue`, `mode`, `status`, `lockedby`, `lockeddate`, `splitfrombatchid`, `copiedfrombatchid`, `privatebatch`, `newbatch`, `uploadstate`
4. Note, that by default, this Server Configuration parameter does not contain any comma-separated values.

For example, in order to group by the *Job name* and the *Step*, the comma-separated list to be used for the `scanapp.client.html.batch_manager_grouped_columns` parameter should be: `jobname, queue`

The outcome should look like the one depicted below:

Batch Manager

Batch ID ↑	Job name	Step	Batch name	Folders	Documents	Pages	Mode
▼ Barcode Job (6)							
▼ Index (3)							
11	Barcode Job	Index	11/30/22 4:11:4...	1	1	3	SUSPENDED
30	Barcode Job	Index	11/30/22 4:12:1...	1	1	7	READY
112	Barcode Job	Index	11/30/22 4:52:...	3	3	16	READY
▼ Scan (2)							
9	Barcode Job	Scan	11/30/22 4:11:3...	1	1	10	SUSPENDED
88	Barcode Job	Scan	11/30/22 4:48:...	3	3	16	SUSPENDED
▼ [Finished] (1)							
2	Barcode Job	[Finished]	11/30/22 4:11:0...	1	1	3	
▼ Email Import (2)							
▼ Index (1)							
41	Email Import	Index	11/30/22 4:13:1...	1	1	18	READY
▼ [Finished] (1)							
63	Email Import	[Finished]	11/30/22 4:13:...	1	1	21	

Total items: 8, last updated: 12:15:25 PM refresh in 14 s auto-refresh every Secs.

Figure 150. Batch Manager with custom grouping

7.3.2. Show/hide the group's header:

Another available customization is the ability to show/hide the group's header at any time.

1. Launch *Thick Client* and open the Server Configuration dialog from the *Tools & Options* menu.
2. Search for the `scanapp.client.html.batch_manager_show_grouping` parameter.
3. The value of this parameter can either be `true` or `false`.
4. Note, that by default, the value of this Server Configuration parameter is set to `true`.

For example, the usage of this Server Configuration parameter is shown below:

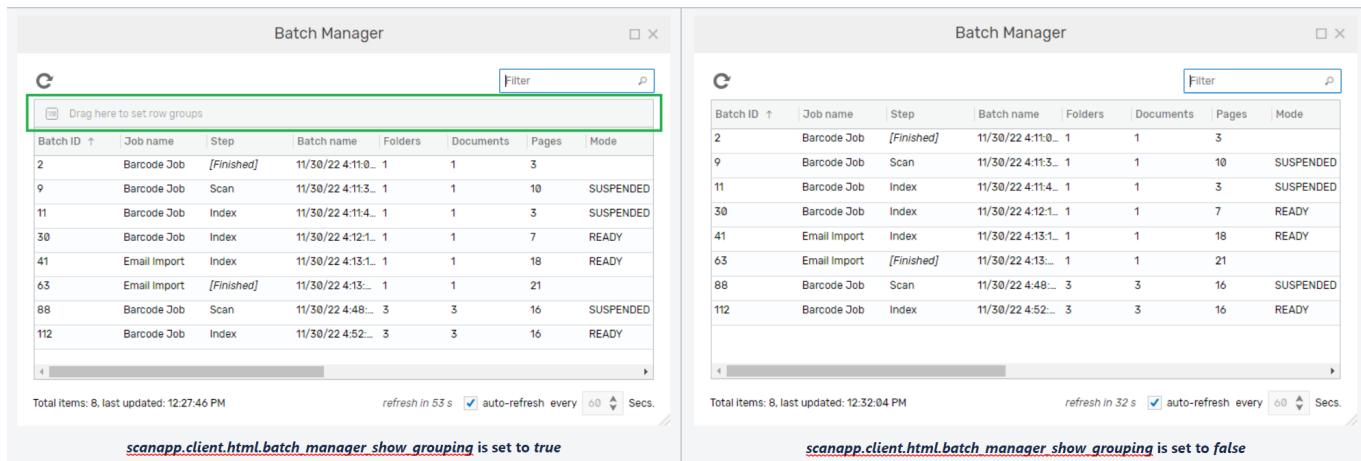


Figure 151. Batch Manager with or without the grouping header

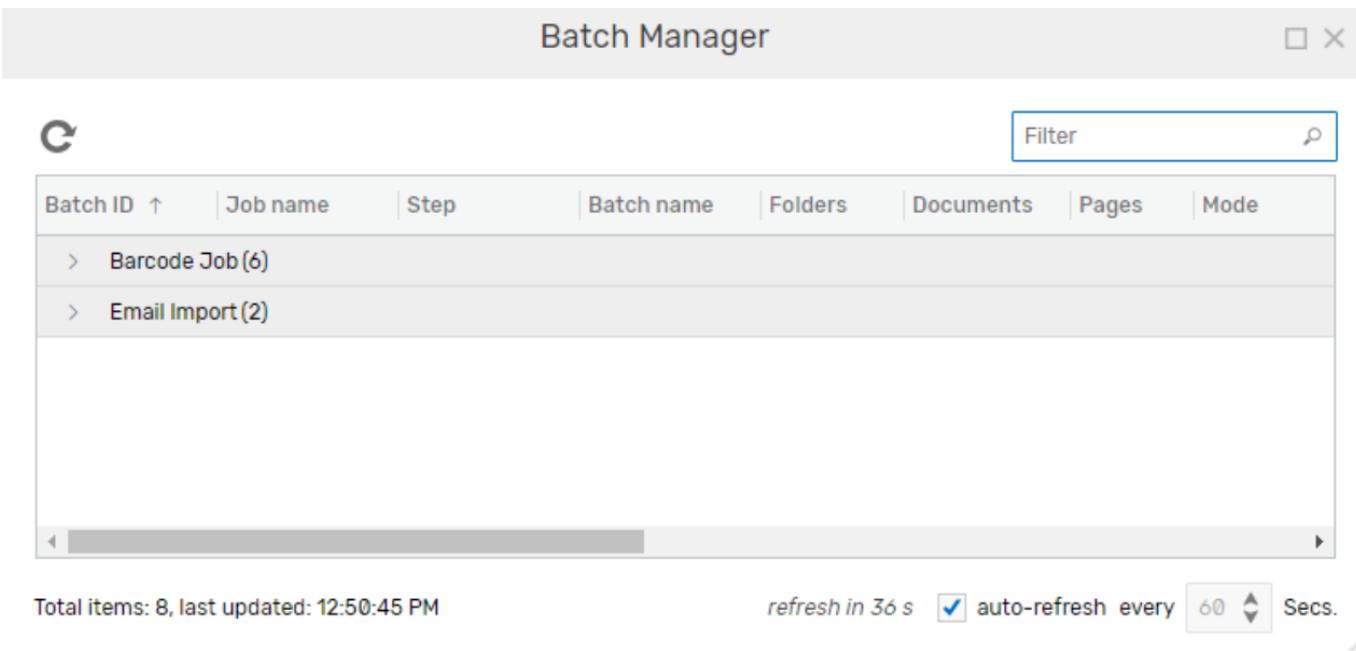
7.3.3. Collapsing the grouped elements of the Batch Manager:

Another available customization is the ability to force the grouped elements of the Batch Manager dialog to be collapsed or not.

1. Launch *Thick Client* and open the Server Configuration dialog from the *Tools & Options* menu.
2. Search for the `scanapp.client.html.batch_manager_groups_collapsed` parameter.
3. The value of this parameter can either be `true` or `false`.
4. Note, that by default, the value of this Server Configuration parameter is set to `false`.

For example, let's suppose that the existing grouping is the one of the "Example 1" shown above → Using the comma-separated values of `jobname`, queue to be used for the `scanapp.client.html.batch_manager_grouped_columns` Server Configuration parameter.

Upon setting the value of the `scanapp.client.html.batch_manager_groups_collapsed` parameter to `true`, the outcome is depicted below:



The screenshot shows the 'Batch Manager' window with a list of items. The items are grouped and collapsed. The first group is 'Barcode Job (6)' and the second is 'Email Import (2)'. The window has a header with 'Batch Manager' and a 'Filter' search bar. Below the header is a table with columns: Batch ID, Job name, Step, Batch name, Folders, Documents, Pages, and Mode. At the bottom, there is a message 'Total items: 8, last updated: 12:50:45 PM', a refresh button 'refresh in 36 s', an auto-refresh checkbox checked with a value of '60', and a 'Secs.' dropdown. A scroll bar is visible on the right side of the list area.

Figure 152. Batch Manager with collapsed grouped elements

7.3.4. Pre-define a subset of Batch Manager columns to be displayed:

Another available customization requires configuring the corresponding Server parameter by using comma-separated values of the names of Batch Manager columns.

1. Launch *Thick Client* and open the Server Configuration dialog from the *Tools & Options* menu.
2. Search for the `scanapp.client.html.batch_manager_columns_list` parameter.
3. Create a comma-separated list using the names of the Batch Manager columns to group by, the available options are the following: `nodetype`, `nodeid`, `nodeorder`, `parentid`, `batchid`, `batchname`, `batchdescription`, `jobid`, `jobname`, `foldercount`, `documentcount`, `pagecount`, `creator`, `creatorgroup`, `createddate`, `priority`, `queue`, `mode`, `status`, `lockedby`, `lockeddate`, `splitfrombatchid`, `copiedfrombatchid`, `privatebatch`, `newbatch`, `uploadstate`
4. Note, that by default, this Server Configuration parameter does not contain any comma-separated values.

7.3.5. Forcing the order of the Batch Manager columns to be the one defined/set:

Another available customization is the ability to lock the order of the Batch Manager columns that are being displayed.

1. Launch *Thick Client* and open the Server Configuration dialog from the *Tools & Options* menu.
2. Search for the `scanapp.client.html.batch_manager_columns_lock_order` parameter.

3. The value of this parameter can either be **true** or **false**.
4. Note, that by default, the value of this Server Configuration parameter is set to **false**.

If the `scanapp.client.html.batch_manager_columns_lock_order` is set to true then the end user will be unable to change the order of the *Batch Manager* columns.

7.3.6. Batch Manager Filters



By default, when no filters are applied, the Batch Manager will always display the most recent batches that were created.

Similar to the Open and Select Next Indexing task windows in both the Thick and the HTML Clients, the Batch Manager is also enhanced with the Filtering mechanism, to be able to display results based on specific conditions. Moreover, using the max-rows option, an Administrator can limit the number of results displayed in the Batch Manager table and optimize the performance.

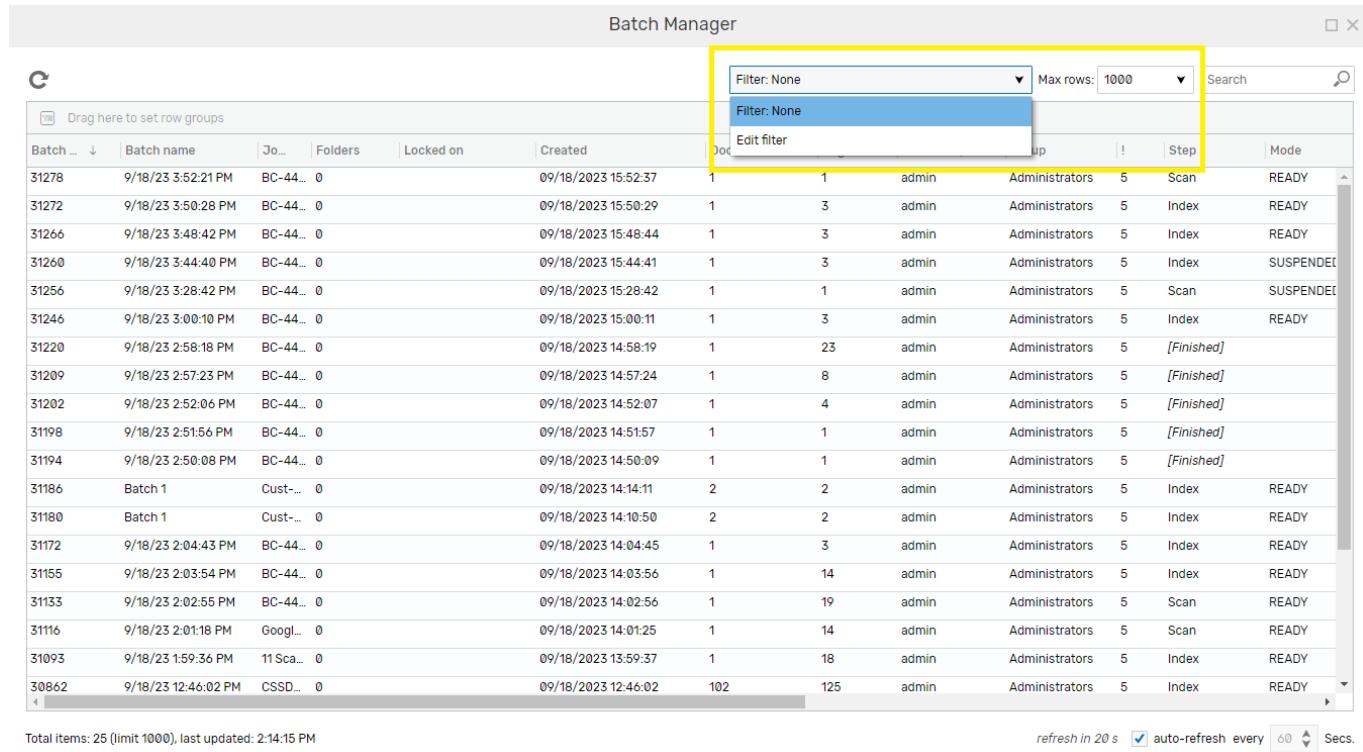
To create or edit a Filter for the Batch Manager, in the Thick Client, or edit the Max rows option, the following options must be configured: The defined filter can be disabled by simply unchecking the Apply filter checkbox.

Type	ID	P...	Folde...	Batch...	Batch name	Desc...	Job ID	Job n...	Fold...	Docu...	Pages	C...	^	Created	!	Step	Mode	Stat...	L...	Lock...	Uplo...	Date 1	Date 2	Date 3
Batch	30757			30757	9/6/23 5:57:59...		527	PNG ...	0	1	1	admin	09/06/2023 15...	5	Index	REA...								
Batch	30763			30763	9/6/23 6:15:45...		527	PNG ...	0	1	1	admin	09/06/2023 16...	5	Index	REA...								
Batch	30767			30767	9/6/23 6:16:39...		527	PNG ...	0	15	15	admin	09/06/2023 16...	5	Index	REA...								
Batch	30799			30799	9/6/23 6:18:35...		528	PNG ...	0	15	15	admin	09/06/2023 16...	5	Index	REA...								
Batch	30831			30831	9/6/23 6:50:24...		528	PNG ...	0	1	1	admin	09/06/2023 18...	5	Scan	SUS...								
Batch	30847			30847	9/18/23 12:19:...		531	BC-4...	1	1	5	admin	09/18/2023 12...	5	Scan	SUS...								
Batch	30862			30862	9/18/23 12:46:...		534	CSV...	0	102	125	admin	09/18/2023 12...	5	Index	REA...								
Batch	31093			31093	9/18/23 1:59:3...		522	11 Sc...	0	1	18	admin	09/18/2023 13...	5	Index	REA...								
Batch	31116			31116	9/18/23 2:01:1...		537	Goo...	0	1	14	admin	09/18/2023 14...	5	Scan	REA...								
Batch	31133			31133	9/18/23 2:02:5...		513	BC-4...	0	1	19	admin	09/18/2023 14...	5	Scan	REA...								
Batch	31155			31155	9/18/23 2:03:5...		538	BC-4...	0	1	14	admin	09/18/2023 14...	5	Index	REA...								
Batch	31172			31172	9/18/23 2:04:4...		539	BC-4...	0	1	3	admin	09/18/2023 14...	5	Index	REA...								
Batch	31180			31180	Batch 1	READY	542	Cust...	0	2	2	admin	09/18/2023 14...	5	Index	REA...								
Batch	31186			31186	Batch 1	READY	542	Cust...	0	2	2	admin	09/18/2023 14...	5	Index	REA...								
Batch	31194			31194	9/18/23 2:50:0...		545	BC-4...	0	1	1	admin	09/18/2023 14...	5	[Fin...	n/a								
Batch	31198			31198	9/18/23 2:51:5...		545	BC-4...	0	1	1	admin	09/18/2023 14...	5	[Fin...	n/a								
Batch	31202			31202	9/18/23 2:52:0...		545	BC-4...	0	1	4	admin	09/18/2023 14...	5	[Fin...	n/a								
Batch	31209			31209	9/18/23 2:57:2...		545	BC-4...	0	1	8	admin	09/18/2023 14...	5	[Fin...	n/a								
Batch	31220			31220	9/18/23 2:58:1...		545	BC-4...	0	1	23	admin	09/18/2023 14...	5	[Fin...	n/a								
Batch	31246			31246	9/18/23 3:00:1...		545	BC-4...	0	1	3	admin	09/18/2023 15...	5	Index	REA...								
Batch	31256			31256	9/18/23 3:28:4...		546	BC-4...	0	1	1	admin	09/18/2023 15...	5	Scan	SUS...								
Batch	31260			31260	9/18/23 3:44:4...		547	BC-4...	0	1	3	admin	09/18/2023 15...	5	Index	SUS...						01/01/1...		
Batch	31266			31266	9/18/23 3:48:4...		548	BC-4...	0	1	3	admin	09/18/2023 15...	5	Index	REA...						01/01/1...	01/01/1...	01/01/1...
Batch	31272			31272	9/18/23 3:50:2...		548	BC-4...	0	1	3	admin	09/18/2023 15...	5	Index	REA...						01/01/1...	01/01/1...	01/01/1...
Batch	31278			31278	9/18/23 3:52:2...		548	BC-4...	0	1	1	admin	09/18/2023 15...	5	Scan	REA...						09/18/2...	09/18/2...	09/18/2...

Figure 153. Define batch manager filters

To create or edit a Filter for the Batch Manager, in the HTML Client, or edit the Max rows option, the fol-

lowing options must be configured:



The screenshot shows the 'Batch Manager' window with a context menu open over a table of batch data. The menu items are 'Filter: None', 'Edit filter', and 'Max rows: 1000'. The 'Edit filter' option is highlighted with a yellow box. The table has columns for Batch ID, Date, Job ID, Folders, Locked on, Created, Doc ID, Step, and Mode. The 'Mode' column shows values like 'READY', 'INDEX', 'SCAN', and 'SUSPENDED'. The 'Edit filter' menu also includes an option to 'refresh in 20s' and 'auto-refresh every 60 Secs.'

Figure 154. Choose batch manager filters

After clicking to Edit the Filters, the window that pops up is exactly the same as the one described in the Task Filters section > Configuring and customizing Task Filters.

The max-rows option can be centrally configured by an Administrator from the `scanapp.client.batch_manager_max_results` Server Configuration Parameter, from the Tools & Options menu > Server Configuration, in the Thick Client. The default value is 1000. An Administrator may set any custom value, to work as an upper limit, or can set the value to -1, for the results to be unlimited.



In general, increasing the complexity of a filter, increases the complexity of the Query to the DB, which makes the filtering task more intensive and thus can result in system instability. Configuring the Max Rows setting to be above 5000 might cause a significant delay to the system and cause it to be unresponsive and is highly discouraged. Requesting batches with custom sorting/filtering options can also impact the responsiveness and smooth operation of the system.

7.4. Queues

Info Input Solution uses the notion of a *Queue* to group all the batches of the system based on their high-level operational state. There are currently 4 distinct queues defined in the system: **SCAN**, **INDEX**,

EXPORT and **FINISH**. In general, batches sequentially move from the first to the last of these queues.

Batches move from one queue to the other automatically, based on the Job's Workflow, the operator's actions and the validity of the index data of the batch. In general, the operator does not really control or choose in a direct way the *Queue* of a batch, but the system selects it as a result of the requested operation.

The general meaning of the queues is:

Queue	Meaning
SCAN	Batches that were just created and are still open in the <i>Client</i> , or have been suspended during a Scan processing step, or moved to Scan Queue as the next Workflow step.
INDEX	Batches in this queue wait to be indexed.
EXPORT	Batches in this queue wait to be exported.
FINISH	Batches in this queue have finished/exited the system and will be deleted soon.

Batches in the **EXPORT** queue are only processed by the *Export Service*.

7.4.1. Where are the batches stored?

All batches are always stored on the server that hosts the *Core Service*. The batch is first uploaded to the *Core Service* when a user *suspends* or *closes* it after the first scan/import.

A user can never delete a batch, with only one exception: if this is a New Batch that has not yet been uploaded to the *Core Service* (or if the user [has the permission](#) to *Delete own batches*). It follows then, that the only way a user can delete a batch is after s/he has scanned it for the first time and before s/he chooses to suspend/close it for the first time.

7.4.2. Batches ownership

By default, the batches of a system have no ownership (with the exception of [private batches](#)). This means that everyone who is allowed access to a batch (security-wise) can open and work on it. This has the effect that any user with access to a set of batches can open any batch s/he wants. So, if **user A** suspends a batch, it is possible for **user B** to resume working on the same batch.

The only way to force the batches to be processed by a single user is to make them *private*. In that case, only the creator of a batch will be able to process it.

7.4.3. How the system understands the "Suspend and Close" user actions

The *Suspend* action is simple and can be performed almost always (with only one exception): the user will temporarily stop what s/he is doing with the intention to resume later. The *Suspend* action carries the idea that the processing that the user needs to do at the specific step is not yet completed. The batch does not change Queue and there are no system checks that could restrict the user from suspending a batch. The only exception, when a user is unable to *Suspend* a batch, is when the *Force indexing on the same workstation option* is enabled in the Job setup options that the Batch belongs to. In this the case, the user is forced to fully index a Batch, before being able to *Close* or *Suspend* it.

The *Close* action means that the user has finished working on the batch and is instructing the system respectively. This is where the system needs to do two things:

- Check any constraints in place (e.g. if an *Index user* tries to close a batch from the **INDEX** queue with empty required fields, the system will not let the user proceed with that action)
- Forward the batch to the next queue, according to the Job Workflow rules.

7.4.4. End user operations

The following operations are available to the end user (from the *Batch menu*):

- *Open batch*: opens a batch from the **SCAN** queue (previously suspended).
- *Select batch to index*: open a batch from the **INDEX** queue.
- *Index next waiting batch*: open the next available batch from the **INDEX** queue.

7.4.5. Customized Get Next Task Definitions

Since version 6.3 it is possible to create custom Task queues, these are shortcuts in the batch menu that open Tasks according to a customized queue. The Custom Definitions are explained in detail in the section [Custom Get Next Task Definitions](#).

7.5. Private Batches

By default, access to batches depends on two factors:

- the *Job* they belong to
- their current state (the [Queue](#) they are in)

Users are [assigned access](#) to Jobs and so, they also have access to all the batches that belong to these Jobs. Moreover, since access to *Scanning* and *Indexing functions* are controlled separately, depending on the current status of a batch, some users may have access to view/edit a batch at a particular point in time, and then lose that ability, as the batch moves to another state.

Info Input Solution provides a secondary mechanism that allows for finer-grain access control to batches, through the function of *Private Batches*. Instead of the standard approach where a batch is a global resource that everyone in a *Group* may have access to, the *Private Batches* approach considers a batch to belong to a single user, the one who created it. No one else has access to this batch, for the lifetime of the batch, except its owner.

7.5.1. Enabling "Private Batches" behavior

The *Private Batches behavior* can be enabled on the *Job* level: from the *Job Setup dialog* → *General job setup properties tab* you can select if you want to enable this behavior:

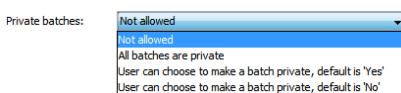


Figure 155. Private batches selection in Job setup dialog

As an administrator, you can choose whether you want all the batches of a specific Job to be private by default, or give the user the ability to choose if s/he wants to make a batch private during creation. If you choose the 3rd or 4th option, then the *Create New Batch dialog* will have an extra option at the bottom-left area that the user may check to make the batch private:



Figure 156. New Batch Dialog: option to make a batch Private

The *Open Batch dialog* (used when opening a suspended batch, or when selecting a batch to index) provides a column to indicate whether a batch is private or not.

7.5.2. Who can view/edit a private batch

By default, *private batches* are only accessible by its creators. There are only two ways to allow someone else access to the private batches of a user:

- there is a permission called *Private batches of other users* that [you can give](#) to a Group/User in order to allow them to have access to the private batches of other users. If a user has this permission, then s/he has access to the private batches of other users, as if they were not private. The other access restrictions (on *Job* and *Function*) still apply. Members of the Administrators group will have this permission by default.

- From the [Batch Manager](#), you can *Unset the private status of a batch* completely, thus making it public again. This is a non-reversible operation that practically makes the specific batch non-private, as if it has never been private in the first place.

8. User Management

The Info Input Solution administrator can design the security model by setting up users and groups, and establishing their roles and security privileges. Moreover, an administrator can review all current active user sessions, and manage the display of *Announcements* to one or more user groups.

8.1. User and Group Administration

The User and Group Administration is accessible to Users who have the *User Administrator* or the *Admin* permission. To access the *User and Group Administration* dialog, go to the *Tools & Options menu* → *User and Group administration...* :

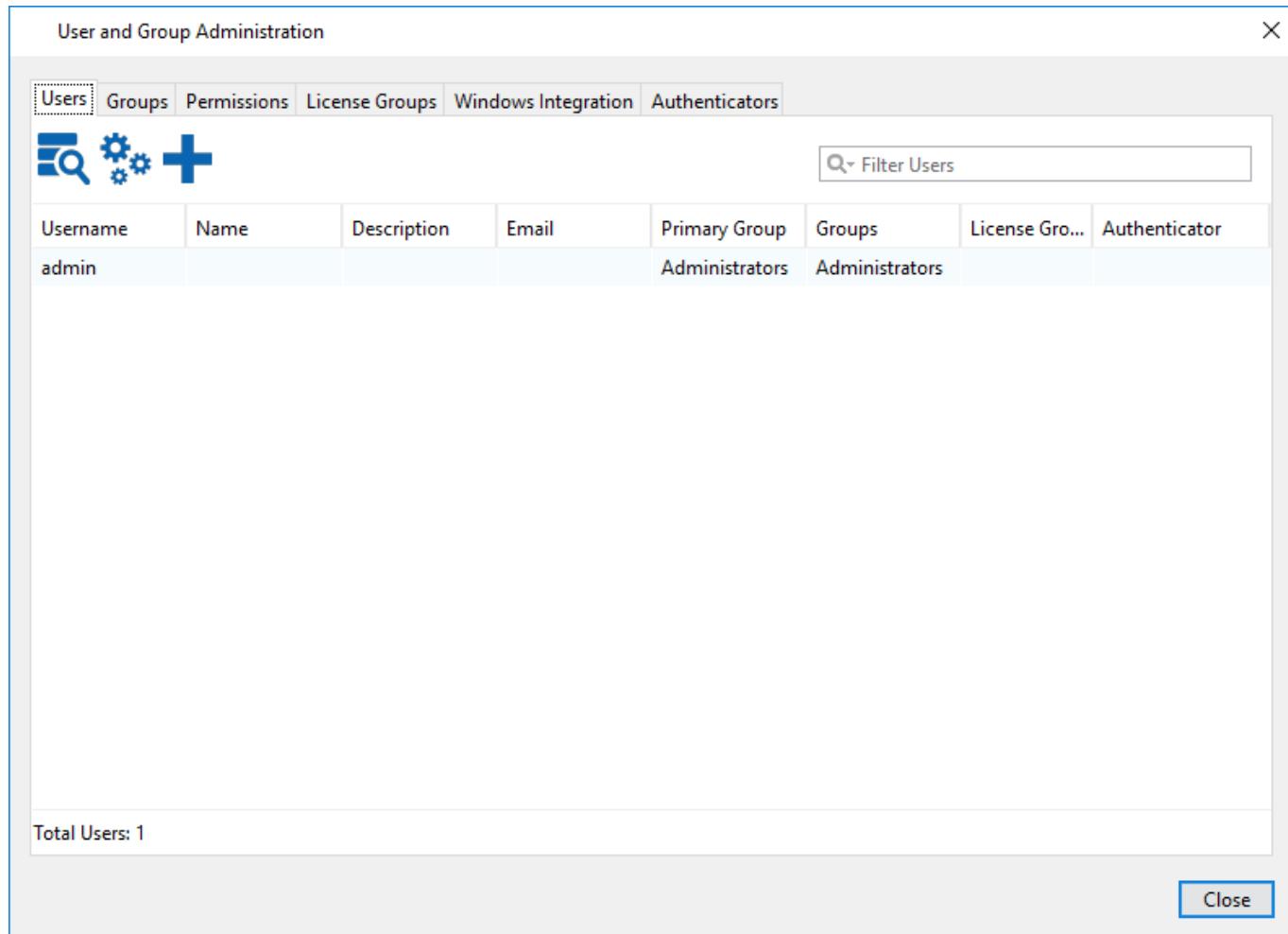


Figure 157. User and Group Administration dialog

The *User and Group Administration dialog* consists of five tabs: *Users*, *Groups*, *License Groups*, *Windows Integration* and *Authenticators*. Each tab displays a table with all the available objects of its kind. All the

tables have facilities for searching, filtering, sorting and grouping of their data. You may select one or more items in a table and bring up the context popup menu to perform certain actions.

All the dialogs that are used in the *User and Group Administration*, save their changes to the *Core Service* when the *Save* button is pressed. Please note that in the User and Group Administration there is no checking for concurrent modifications. So, if more than one persons make changes at the same time, it is possible that one person's changes will be overwritten by another person.

8.1.1. Users

To create a new User, press the *Create new User* button in the *Users* tab.

New User X

Username: *	batch_admin
Password: *	*****
Confirm password: *	*****
Full name:	
Description:	batch administrator
Email:	
License Group:	[No License Group] ▼
Primary Group: *	Unit_Administrators ▼

Fields marked with * are required

Member of the following Groups: grid icon *

<input type="checkbox"/> Administrators
<input checked="" type="checkbox"/> Unit_Administrators
<input type="checkbox"/> global

Filter Groups Save Cancel

Figure 158. New User dialog

If a system only has the default *System Authenticator*, then all newly created Users are automatically assigned to the *System Authenticator*. In this case the Authenticator combo box is not visible. If the system has more than one *Authenticators*, then you need to select the appropriate *Authenticator* for the new User.

The username must be at least 2 characters long and it must be unique within the selected *Authenticator*.

The password field is required when the selected *Authenticator* is the *System Authenticator*. Users that belong to other Authenticators do not have a password within Info Input Solution.

Info Input Solution does not impose any restrictions on the password, except that it cannot be empty. If you wish to impose certain restrictions for the password, then you need to implement the javascript function `isPasswordValid` in the *Global script*.

Normally an administrator creates Users using some default password and the Users are able to change their password when they login into Info Input Solution. The administrator can also reset a User's password.

Every User must be a member of at least one Group. When an external *Authenticator* (like a Single sign-on service or an LDAP service) is used to control access to Info Input Solution, then the list of *Groups* that the authenticated *User* is a member of is decided as follows:

- If the Authenticator has the *Auto-assign Groups* option enabled, then the User is assigned to the *Associated Groups* of the *Authenticator*. In this case, the administrator cannot change the Group assignments for the User.
- If the Authenticator has the *Auto-assign Groups* option disabled, then the User can be assigned to any of the available *Groups*. When a new User is created, the *Authenticator's Associated Groups* (if any exist) are used in order to preselect the new User's Groups, but the administrator may change the selection.

8.1.2. Editing multiple Users

The Users tab allows editing of multiple Users at the same time. It is possible to select multiple Users and press the *Properties* context menu item.

Properties for Users: spanish, german, arabic, chinese, english X

Username: *	spanish, german, arabic, chinese, english
Full name:	[Multiple values]
Description:	[Multiple values]
Email:	[Multiple values]
License Group:	[No License Group] ▼
Primary Group: *	global ▼

Fields marked with * are required

Member of the following Groups:

Administrators
 Unit_Administrators
 global

grid icon *

Filter Groups

Save Cancel

Figure 159. User dialog while editing multiple Users

This is useful if you want to assign or remove *Groups* from multiple *Users*. You can also change the *primary Group* and the *License Group* to multiple *Users*.

8.1.3. Groups

Users can be assigned to different *Groups* and every *Group* has a set of *Permissions*. This allows the administrator to manage the roles that each User has.

The security subsystem of Info Input Solution works as follows:

- *Permissions* are used to allow/disallow access to *functions* or *Jobs*
- *Permissions* are assigned to *Groups*. Each *Group* may have one or more *Permissions*. The same *Permission* can be assigned multiple times to different *Groups*.
- *Users* can be members of one or more *Groups*. *Users* inherit all the *Permissions* of the *Groups* they are members of.

When a *User* logs in, the system finds out all the *Groups* s/he belongs to, and from there it finds out all the *Permissions* s/he has: this list of permissions is used from then on, both on the *Client* and on the *Core Service*, to authorize what the user can or cannot do.

8.1.4. Permissions

Info Input Solution supports permissions on two different levels:

- on the *function* level: these types of permissions enable access to a specific function of the system (for example *Indexing*)
- on the *object* level: these types of permission enable access to a specific object of the system (e.g. a specific *Job*)

From the *Groups* tab you can *Create/Edit Groups* and assign them with selected permissions: the *Functions* lists refers to function-level permissions and the *Objects* lists provides object-level permissions. For example, if you create a *Group* with the permissions *Functions*→*Scan*, *Objects*→*JobA*, then a user belonging to this *Group* will only be able to create batches of this *Job* and scan images. Notice the special Object permission *All jobs* that gives one access to all the *Jobs* of the system.

Specifically, the function level permissions are:

Administrator

provides administration privileges to the entire system; this permission gives access to every function of the system. In other words, the Administrator permissions is a complete set of all listed function level permissions.

User Administrator

allows users to create/edit *Users*, *Groups*, *License Groups* and *Authenticators*.

Job Administrator

allows uses to create/edit *Jobs* and all the shared entities (like *Field Types*, *Export Destinations* etc) that are using in *Job* definition.

Scan

allows users to use the *Scanning functions* of Info Input Solution. You need to select at least one *Job* from the *Objects list*.

Index

allows users to use the *Indexing functions* of Info Input Solution. You need to select at least one *Job* from the *Objects list*.

Batch Manager Access

allows access to the [Batch Manager](#); the *Batch Manager* can perform several administrative tasks.

Batch Manager-Change Mode

allows users to change/update the Mode of selected Batch(es).

Batch Manager-Change Priority

allows users to change the priority of any Batch(es) by setting it higher or lower.

Batch Manager-Change privacy status

allows users to change the Private Status of any Batch(es). Upon a user selects to remove the Privacy Status of any particular Batch(es), then, this/these Batch(es) will be visible to all users that have access to the Job(s) that this/these Batch(es) belong(s) to.

Batch Manager-Change properties

allows users to change the properties of any Batch Node(s). E.g. a user may be allowed to change the Batch Description, the Batch notes, etc.

Batch Manager-Change step

allows the users to manually change the Workflow Step of any selected Batch(es).

Batch Manager-Create Batch copy

allows the users to be able to create Batch copies of a particular Batch(es) and provides several options on how the new Batches will be created.

Batch Manager-Delete

allows the users to be able to delete Batches from within the Batch Manager.

Batch Manager-Promote Batch to latest Job version

if any modifications are applied to a Job Setup, then, this function-level permission will allow the users to be able to promote an existing Batch that was previously created, to use the latest version of this particular Job.

Create Batch

allows users to create a new Batch.

Work Offline

enables users to launch the *Thick Client* and work offline, when the connection to the *Core Service* cannot be established. See the [Offline Scanning](#) topic for details.

- *Private Batches of other Users*: allows access to view private batches of other users. See the [Private Batches topic](#) for how this permission can be useful.

Delete own Batches

allows users to delete their own batches. Normally, once a batch has been committed to the *Core Service*, a user cannot delete it: this permission allows a user to delete those batches s/he has created, even after they have been committed to the *Core Service*.

Create global announcement

See the [Announcements](#) topic for details.

Create group announcement

See the [Announcements](#) topic for details.

Overwrite Job's scan Profile

allows users to use his/her [local Scan Profiles](#), even for Jobs that have [Global Scan Profiles](#) assigned.

Overwrite Job's scan Image Mode

for Jobs that have [scan Image Mode\(s\) defined](#), it allows users to overwrite the Job level setting.

Overwrite Job's Scan Page Mode

for Jobs that have [scan Page Mode defined](#), it allows users to overwrite the Job level setting.

Overwrite Job's scan resolution

for Jobs that have [scan Resolution defined](#), it allows users to overwrite the Job level setting.

8.1.5. License Groups

License Groups are named groups that may contain one or more users. Each *License Group* has a *Maximum number of users* that allows to be simultaneously logged in at any moment. A user may optionally belong to a *License Group*. The system will not allow more than the maximum allowed number of users that belong to the same *License Group* to be logged in at any moment.

You can use *License Groups* to divide the concurrent licenses of your system to different business units.

For example, let's assume you have a system with **20** concurrent user licenses and you have in total **80** named user accounts that can login at any time to the system. Obviously only **1 out of 4** of the users can use the system at the same time. Also, let's assume that you need to be sure that there are at least **4** licenses that are not being shared among the **80** users, but are kept exclusively to be shared by **6 super-users**: you can create a *License Group* named *Normal Users License Group* with **16 Max Users** allowed and assign this group to *all but the 6 super-users*: now, every time any of the **74** normal users logs in, they will take a license from the *Normal Users License Groups* (e.g. one of the **16**), whereas the remaining **4** licenses will not be used. If one of the **6** super-users logs in, and since these do not belong to a *License Group*, the system will take one of the remaining **4** licenses.

You *create and edit License Groups* from the *License Groups tab*. You assign a user to a *License group* from the *Users tab* → *License groups list*. You assign a whole *Group* to a *License Group* from the *Groups tab* → *License group list*.

8.1.6. Windows Integration

Info Input Solution *User* permissions can be integrated with Windows group policy, so Windows users can be authenticated without providing Info Input Solution *User* credentials. Note that *Windows Integration* does not apply to the *HTML Client*, since the Web browsers do not allow Web Applications to gain access to the underline OS.

When Info Input Solution Login popup dialog is showed and "Windows User Login Credentials" option is used, Windows user domain and group membership are provided to Info Input Solution. If any of these are matched according to Info Input Solution *Windows Authenticator* configuration then the *User* successfully logs in and grants Info Input Solution *Group* permissions defined on *Windows Authenticator*.

Here is an example that the Windows Authenticator will assign *Index Group* to Windows Users with membership in a custom Windows Group (SID: S-1-5-21-3985899430-2012974364-4046225244-1004)

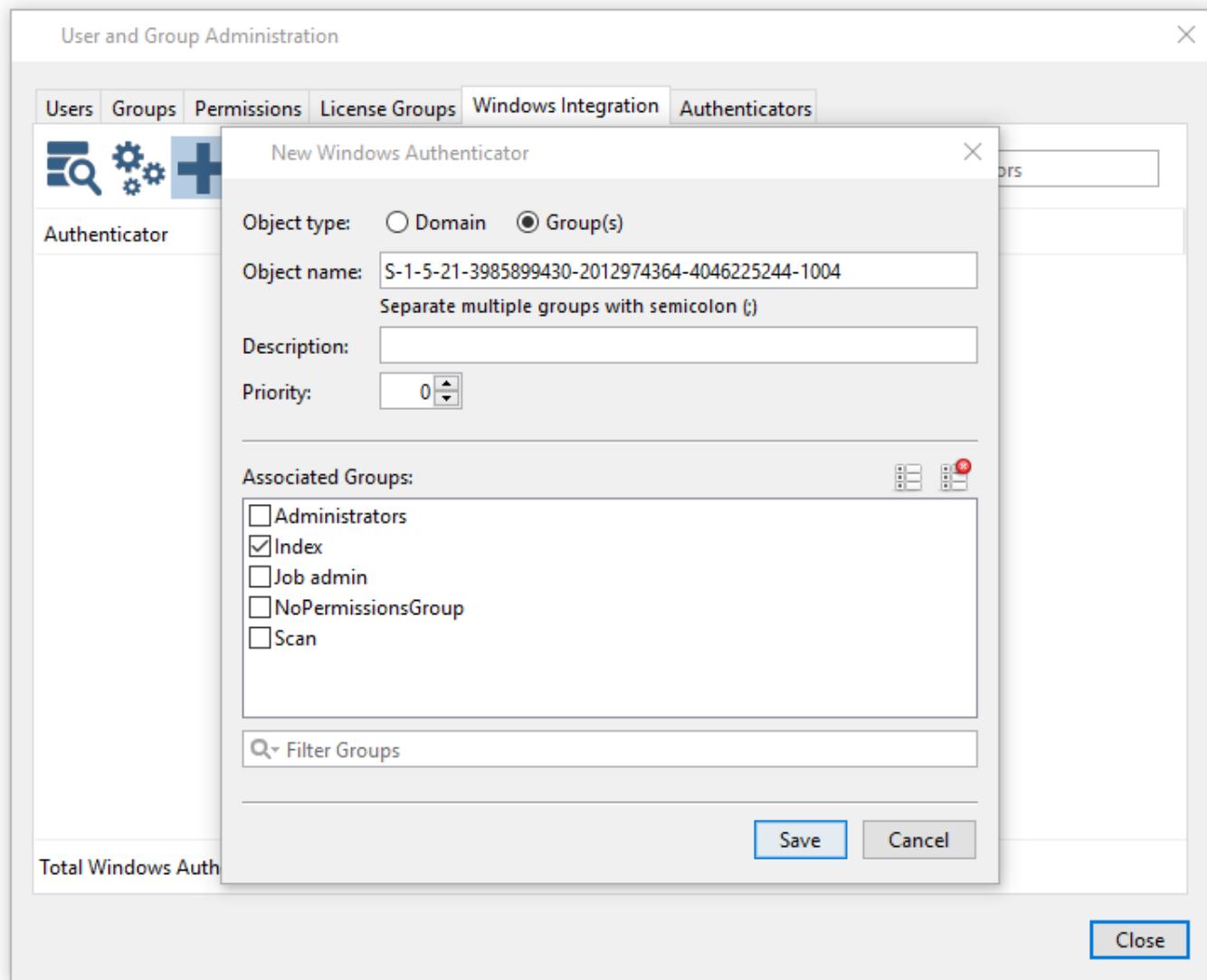


Figure 160. User dialog while editing multiple Users

If more than one Info Input Solution *Groups* are selected, then all permissions provided by these *Groups* will be assigned to the authenticated user. In the *Object name* field, multiple Windows group SIDs can be added separated by a semicolon (;). Windows users that are members of all these groups will successfully login.

For Windows users that belong to more than one Windows groups, multiple Windows *Authenticators* can be configured and a *Priority* value can be defined for each of them. During login, all Windows *Authenticators* will be used sequentially from lower to higher *Priority* values, to authenticate the *User*. Info Input Solution permissions will be granted according to the first Windows *Authenticator* that will successfully authenticate the user.

It is also possible to authenticate users that belong to a common Windows domain or group and it is needed to provide different Info Input Solution *Group* permissions according to each *User Windows*

group membership.

This is performed in two steps:

1. Create a *Windows Authenticator* that will allow *Users* to successfully login when they belong to the Windows domain/group:

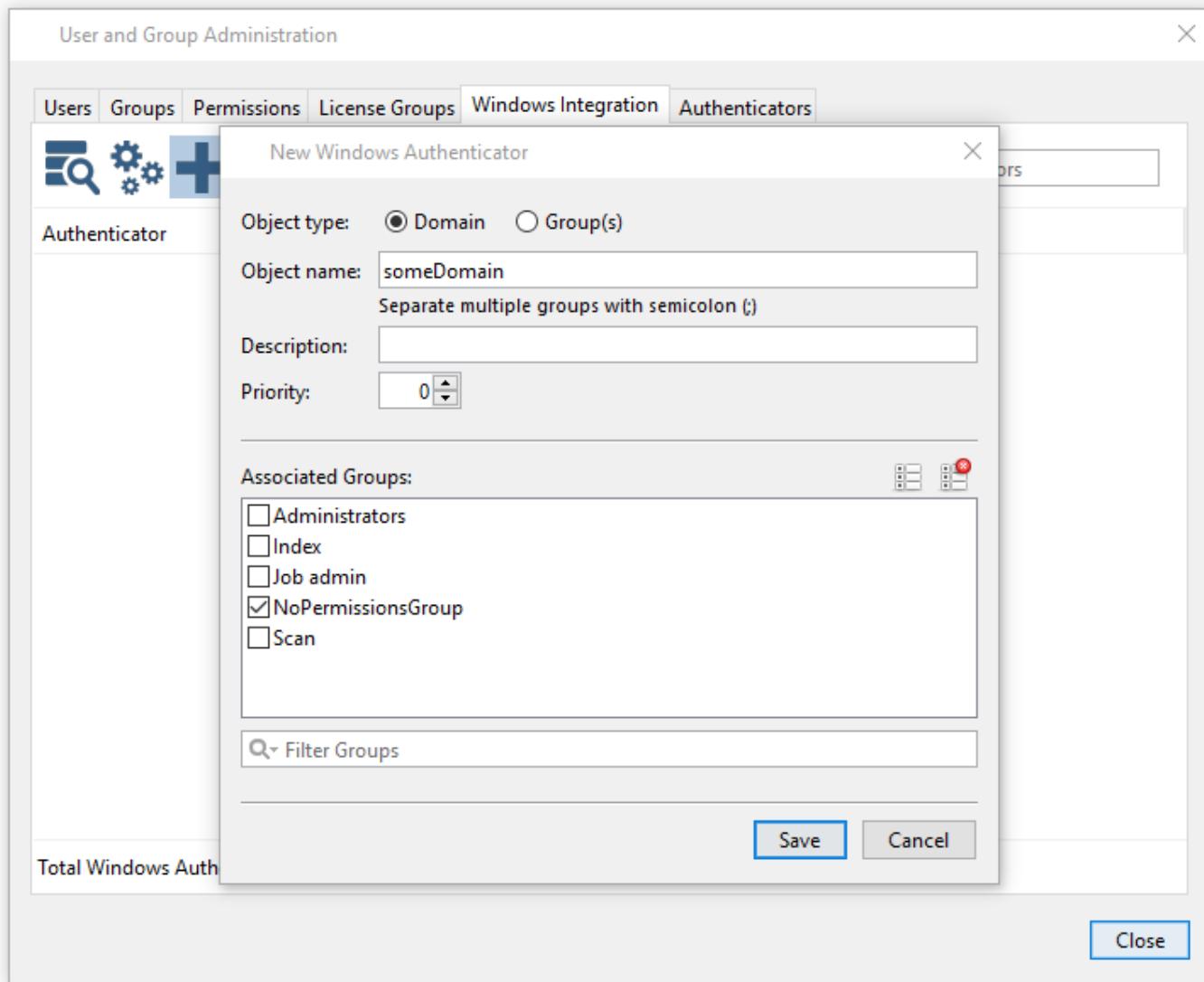


Figure 161. Windows Domain Authenticator

Info Input Solution *Group* with no permissions is used to prevent users from granting additional permissions.

1. Assign Windows group SID value to "External Matching ID" field in Info Input Solution *Group*:

Properties for Group Index

Name:	Index
Description:	
License Group:	[No License Group]
External matching ID:	S-1-5-21-3985899430-2012974364-4046225244-1004

Functions:

- Administrator
- Batch Manager
- Create Batch
- Create global announcement
- Create group announcement
- Delete own Batches
- Index
- Job Administrator
- Override Job's Scan Profile
- Override Job's scan Image Mode
- Override Job's scan Page Mode
- Override Job's scan resolution
- Private Batches of other Users
- Scan
- User Administrator
- Work Offline

Objects:

- All Jobs
- tesf

Filter Functions

Filter Objects

Save Cancel

Figure 162. Assign Windows SID to application Group

8.1.6.1. Authenticators

The users can also login to Info Input Solution application using a domain account and be authenticated by a third party *Authenticator*.

A *User* will successfully login, if it is authenticated successfully by the third party *Authenticator* and at least one Info Input Solution *Group* is assigned. Info Input Solution *Group* membership defines *User* permissions inside Info Input Solution.

When a successful domain authentication is performed, a list of domain group names that the *User* is member of, is returned. Then the Info Input Solution *Group* that is matched with each returned domain

group name, is assigned to the authenticated Info Input Solution *User* and the *User* logins.

At least one default Info Input Solution Group should be defined, in case no Info Input Solution and domain account *Group* names are matched. For these cases, it is recommended to create only a blank (without permissions) Info Input Solution *Group* that will be assigned to the authenticated *User*.

When a user successfully logins, it has access to the superset of matched Group(s) and the default Info Input Solution *Group(s)* permissions.

Here is how a Microsoft Active Directory (AD) *Authenticator* can be setup:

You can edit one or more AD Authenticators, in the Properties popup dialog.

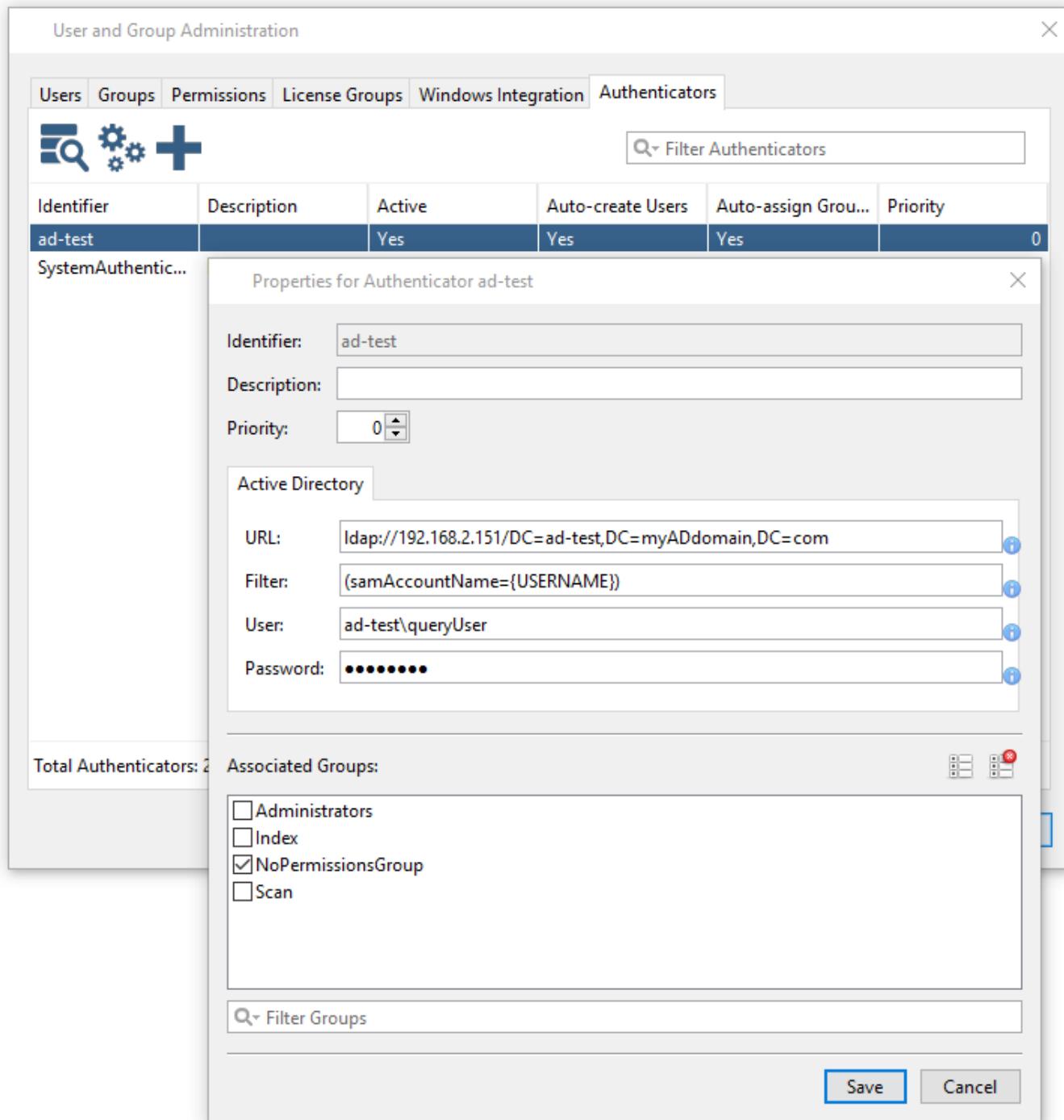


Figure 163. Authenticator dialog

Identifier

The domain identifier. This is what the users will type as a prefix before backslash ("\") in the user-name field on the Login dialog

URL

The Active Directory URL. For example: `ldap://<Server name/ url/ ip>/DC=ad-test,DC=yourDomain,DC=com`. In case the Active Directory server has the LDAP Binding and LDAP signing enabled the following URL structure should be used `ldaps://<FQDN>/DC=ad-test, DC=yourDomain, DC=com`. The FQDN should be the same as the one that is set in the certificate that is used in AD

Filter

This is the expression that will be used to find the user in the domain. For example the following will search for objects that the `samAccountName` will match the username field from the log in dialog: `samAccountName={USERNAME}`

User, Password

The username and the password of a user that can execute queries in the Active Directory.

Associated Groups

The *Authenticator* will search for AD groups that their names match Info Input Solution *Group* names. If none of the groups match, a superset of checked Associated Groups will be used. It is recommended to create and check only a blank (without permissions) Info Input Solution Group.

8.1.6.2. SSO Integration

When Integrating with a third party SSO mechanism, *User*'s account profile information can be directly used to control authorization in Info Input Solution. So Info Input Solution can be member of an Application group that *User* will provide credentials only once and will have access to all of them.

It is common in SSO systems, an agent in the form of a reverse proxy to be placed "in front" of Web Applications. In Info Input Solution SSO Integration, this agent authenticates the user, if not already authenticated, and then forwards the request to the Thick Client, passing *User*'s identity information, in the form of HTTP headers.

It is necessary to decide which *User* Identity pieces to transfer. Info Input Solution requires *Username* uniqueness. The *Username* must be unique for the SSO system and the associated internal *Authenticator* (*Authenticator Configuration* paragraph below). If *User* authorization is managed centrally, through the user directory, then the information reflecting the user's authorization (e.g. domain group membership) should also be captured and transferred.

The HTTP request that also includes HTTP headers *User* identity information will be handled by `<root_installation_dir>/client/sso/index.jsp`. An SSOPayload Data Transfer Object will be created and passed to the *Thick Client* as an encrypted XML payload. Info Input Solution installation comes along with a SiteMinder `/client/sso/index.jsp` example page.

When the *Thick Client* need to launch, then SSO agent should call the following URL: <http://<server->

Name>:<serverPort>/client/sso

SSO Integration is also available for the *HTML Client*. To enable it, modify <root_installation_dir>/client-html/index.jsp and set "UseSSO" parameter to true

```
var ithConfig = {  
    .  
    .  
    .  
    "UseSSO": true  
};
```

Then the SSO agent should call the following URL and the Apache Tomcat service forwards the request to /client/sso/index.jsp for processing: <http://<serverName>:<serverPort>/client-html>

8.1.6.3. User identity - JAAS Subject / Principals

As *Authentication and Authorization* section describes, Info Input Solution uses *Java Authentication and Authorization Service* (JAAS) to authenticate users with various Authentication Services / Directories.

In *JAAS*, a *Subject* is an object describing *User*'s identity: username, authentication domain, domain group memberships, privileges and permissions as well as full name or email address. All this information is expressed as *Principal* objects of various types.

A *Subject / Principals* combination is the form of *User* identity information that Info Input Solution understands and obtains from *JAAS* when a *User* logs in, using the *Login* dialog. Info Input Solution *SSO Integration* is a mechanism that obtains a *User* identity from the SSO agent and translates it into a *Subject / Principals* combination. The first thing Info Input Solution *SSO Integration* has to do is to capture this information and pass it to the Client.

Info Input Solution can then recognize, authorize and give access to the *User*, as if the *User* was an internal Info Input Solution *User*. Moreover, *SSO Integration* can sit alongside Info Input Solution internal *User Authentication* mechanism. All of these *Users* can then be authorized uniformly from within the *User and Group Administration* GUI.

The *JAAS login helper class* that will be used to translate *User* identity to a *Subject / Principals* combination should be included in the *SSOPayload* Data Transfer Object transmitted to the Client. Since this *Subject / Principals* combination may differ between SSO systems and scenarios, some custom logic is needed to perform the actual work. When Client finds *login_helper* in its list of parameters, it assumes its value is the fully qualified name of a Java class implementing *LoginHelper* interface and switches to SSO

integration mode.

In SSO integration mode, the Client constructs an instance of the class named in the *login_helper* parameter and calls its `getSubject()` method to acquire a *Subject*. It then sends this Subject to the *Core Service*, to conclude the user log-in process. The code sample below shows how a *LoginHelper* implementation could construct a *Subject*, using the information passed to the Client in the HTML page:

```
...
@Override public
Subject getSubject(Environment e) throws LoginHelperException {SSO_SHOW_LOGIN_ON_ERROR
    final String username = e.getParameter("ssoUserName");
    if (username == null || username.isEmpty())
        throw new LoginHelperException("Cannot find ssoUserName in environment");
    Subject subj = new Subject();
    subj.getPrincipals().add(new UsernamePrincipal(username));
    subj.setReadOnly();
    return subj;
}
```

8.1.6.4. Processing the Subject on the Core Service

When *Core Service* receives a Subject from the *Client*, it performs the following tasks:

Match the Subject to an Authenticator: the *Core Service* considers every *Authenticator* in its database and attempts to match its Principal Conditions against the Principals in the Subject. Only when all Principal Conditions are satisfied for an *Authenticator* does the *Core Service* conclude that it matches with the Subject. If the *Core Service* fails to match the Subject with an *Authenticator*, the entire *User* login process fails.

Find an existing or create a new User object: regardless of the authentication source (Info Input Solution itself, or an external service like AD or an SSO system), Info Input Solution needs to have an internal *User* object for an identity. Depending on the *Authenticator*'s configuration, Info Input Solution can either create such an object, if it does not already exist, or require the one has been created by the *Administrator*. These two options correspond to the two options available for granting Info Input Solution access to *Users* (centrally or from within Info Input Solution).

Assign Info Input Solution Groups to the User object: if *Authenticator Configuration* allows, the *Core Service* will refresh *User*'s Group memberships, associating the *User* with both the *Groups* statically associated with the Authenticator and those dynamically provided at login time. The latter include any *Groups*

provided by the *SSO Integration*. On the contrary, *Authenticator Configuration* may impose that *User Group* memberships are not refreshed on every login and once done by the administrator will always apply.

8.1.6.5. Authenticator Configuration

For every task the *Core Service* performs while processing a client-provided *Subject*, its behavior is imposed by the *Authenticator Configuration*. This configuration is a set of records in the Info Input Solution database. The primary record in table *AuthenticatorMatchers* describes the *Authenticator* itself:

id

The numerical ID of the database record. Must be provided on insertion for Oracle (using sequence *AuthenticatorMatchers_seq*), must not be provided for SQL Server.

matcherType

Has to be "*BaseAuthMatcher*"

version: Should be initially set to 0 (zero)

uuid, identifier

A unique name that identifies this *Authenticator*

namingPrincipal

The fully qualified name of the Principal class, an instance of which must exist in the *Subject*, which will contain the username of the user.

active

Must be "Y"

priority

The priority of an *Authenticator* compared with the other *Authenticators* in the database. *Priority* value affects *Authenticator*'s ordering that the *Core Service* will use when iterating over all, trying to find the matching *Authenticator* for a given *Subject*. Smaller numbers means greater *priority*.

createUsers

"Y" or "N". Imposes whether the *Core Service* should create a new *User* object for a user that accesses the Info Input Solution for the first time. If this is "N" and a *User* object does not already exist for the user, the log-in will fail for the user.

refreshGroups

"Y" or "N". Imposes whether the *Core Service* should refresh the Group memberships for the *User*, or not. To have dynamic Group memberships from the *SSO Integration*, this must be "Y".

description

A short description about the *SSO Authenticator*

The Subject-Authenticator matching conditions are configured for an *Authenticator*, by inserting a record into table *AuthenticatorMatcherConditions* for every matching condition. A condition is essentially a check on the type (concrete sub-class of Principal) and optionally the value of the Principal object. There can be more than one conditions for an *Authenticator*, in which case all must be satisfied for the match to succeed. A condition record consists of the following properties:

authenticatorMatcherId

The numerical ID of the *Authenticator*'s record in the *AuthenticatorMatchers* table

listIndex

A zero-based index for each condition in the list of conditions for a given *Authenticator*. The value dictates the order the conditions are checked. There shouldn't be any gaps in the list for an *Authenticator*.

principalClass

The fully qualified name of the Principal sub-class

property

The JavaBean property of the Principal sub-class, the value of which should be considered for the match.

operator

The comparison to perform between the value of the condition and the value obtained from the Principal object, one of "EQUALS", "STARTS_WITH", "ENDS_WITH", "CONTAINS" and "REGEX".

value

The condition value

For a matching condition, the property, operator and value can be null, in which case the condition is simply a check on the type of Principal: the Subject has to have a Principal of the specified class for the condition to be satisfied.

An example *Authenticator Configuration* is the following:

- Record in *AuthenticatorMatchers*:
- *identifier*: 'OAM-SSO'
- *namingPrincipal*: 'scanclient.login.sample.UsernamePrincipal'
- *createUsers*: 'Y'

- *refreshGroups*: 'Y'
- Record in *AuthenticatorMatcherConditions*:
- *listIndex*: 0
- *principalClass*: 'scancient.login.sample.UsernamePrincipal'
- *property, operator, value*: null

Here is how a SiteMinder *Authenticator Configuration* can be created:

```
INSERT INTO "ITSCH"."AuthenticatorMatchers" (matcherType, version, uuid, identifier, namingPrincipal, active, priority, createUsers, refreshGroups, description)
```

```
VALUES ('BaseAuthMatcher', 0, 'SiteMinder', 'SiteMinder', 'scancient.login.siteminder.SMUserPrincipal', 'Y', 2, 'N', 'N', 'SiteMinder');
```

```
INSERT INTO "ITSCH"."AuthenticatorMatcherConditions" (authenticatorMatcherId, principalClass, property, operator, value, listIndex)
```

```
SELECT id, 'scancient.login.siteminder.SMUserPrincipal', null, null, null, 0 FROM "ITSCH"."AuthenticatorMatchers" WHERE identifier = 'SiteMinder';
```

8.2. Authentication and Authorization

Info Input Solution's *User Authentication and Authorization Mechanism* was designed to easily integrate with central Authentication Services and Directories, utilizing industry-standard technologies:

- It utilizes *Java Authentication and Authorization Service* (JAAS) to authenticate users with various Authentication Services / Directories.
- It is compatible with all JAAS-compliant *Login Modules*
- It does authorization via *JAAS-provided Principals* and additive (logical AND) checks on their types and attributes. Attribute checks with various operators are supported (equals, starts-with, ends-with, contains, matches regex).
- Users are authorized by mapping *Authentication Services* to one or more *User Groups*.
- The configuration is done via the Client, so there is no need for editing configuration files or Database tables (currently only for *Active Directory / Windows integration*).

The following diagram illustrates the architecture of the mechanism.

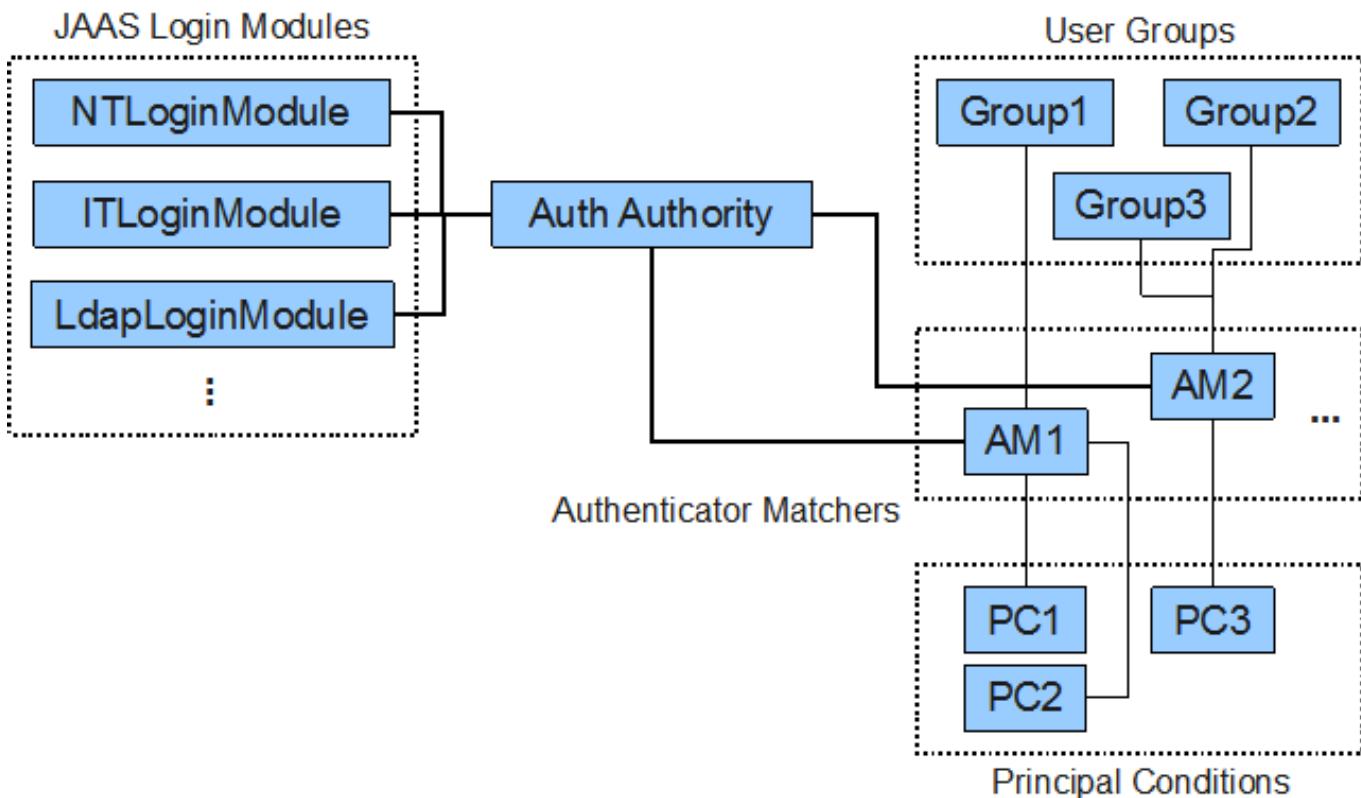


Figure 164. Authentication Mechanism

8.2.1. JAAS and Login Modules

JAAS implements a Java version of the standard *Pluggable Authentication Module* (PAM) framework, which is used to authenticate users in various operating systems, such as AIX, HP-UX, GNU/Linux, Mac OS X, Solaris and some variants of BSD. JAAS, like PAM, abstracts the action of authenticating a user by hiding the implementations of the various authentication protocols and providing applications with a single, implementation-agnostic API. JAAS makes it possible for an application to authenticate users from a variety of Authentication Services at the same time.

Login Modules are the Java implementations of the various authentication protocols. There are modules to integrate with various Authentication Services and Directories, like *LDAP*, *Active Directory* and *Kerberos*, and modules to enable Single-Sign-On in conjunction with Operating Systems, like Microsoft Windows and variants of UNIX. It is also possible to implement custom *Login Modules* to integrate with any custom Authentication Service or Single-Sign-On infrastructure.

Info Input Solution fully utilizes JAAS to be able to authenticate users with its own Database, authenticate users with external Authentication Services, or retrieve user identity from Single-Sign-On infrastructures, all at the same time and in a fully configurable way.

8.2.2. JAAS Entities: Subjects and Principal

JAAS encapsulates all Identity information about an authenticated entity in an object called a *Subject*. A *Subject* contains a set of *Principals*, which are the actual bits of *Identity information*, for example *Windows Domain* or *Group name*, entity's system username, person's actual name, etc. These bits of Identity information can be queried and examined to control access to and authorization of the entity to the system.

Apart from the actual Identity information they contain, *Principal objects* can be used to determine the type of *Login Module* (or *Authentication Service*) that generated them, based on their Java type. For example, a *Principal* of Java type `com.sun.security.auth.LdapPrincipal` can only be generated by *LdapLoginModule*, thus indicating that it was a *LDAP Directory* that authenticated the related entity.

It is these properties of *Principal objects* that Info Input Solution utilizes to control access and authorization of its users.

8.2.3. Authenticator Matchers and Principal Conditions

Authorization in Info Input Solution is based on user *Groups*. *Jobs* are made visible to user *Groups* and permissions are granted to user *Groups*. Each internal Info Input Solution *User* (i.e. users defined within Info Input Solution and authenticated by it directly) is assigned to one or more *Groups*, but this cannot be done with unknown, externally-authenticated users.

Authenticator Matchers bind external *Authentication Services* or *Single-Sign-On infrastructures* (collectively called *Authenticators*) with Info Input Solution user *Groups* to do authorization of all its users in a single, unified way.

Authenticator Matchers examine each incoming *Subject* (user) to determine the Authenticator that provided it with the help of *Principal Conditions*. These are additive (logical AND-related) checks on the Subject's Principals that can examine both its Java type and any other attributes. A Principal's attribute can be examined for equality with a certain value, whether it starts-with or ends-with a certain value, contains a value or matches a regular expression.

Examples of Authenticator Matchers' sets of Principal Conditions:

- Principal Type is `com.sun.security.auth.NTDomainPrincipal` AND *Name* attribute is equal to `WINDO-MAIN`
- Principal Type is `com.sun.security.auth.NTSidGroupPrincipal` AND *Name* attribute is equal to `123-456-789`
- Principal Type is `com.sun.security.auth.LdapPrincipal` AND *Name* attribute contains `OU=SomeDept, O=MyOrg`

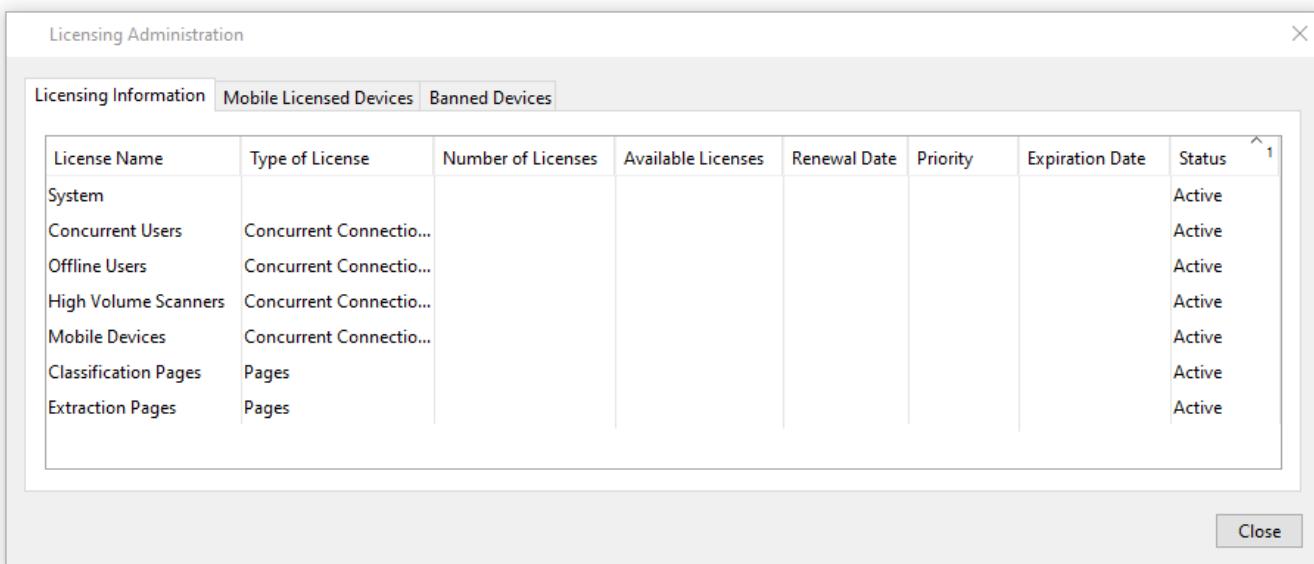
The first example defines an *Active Directory Authenticator*, which authenticates all users of the *Windows Domain* **WINDOMAIN**. The second example defines again an *Active Directory Authenticator*, which authenticates users that belong to *Windows Group* with *SID* **123-456-789**.

The third example defines a *LDAP Authenticator* which authenticates users whose *Distinguished Names* (DNs) contain the specified string, that is they are employees of organization **MyOrg**. and work in unit **SomeDept**.

8.3. Licensing Administration

The *Licensing Administration* module is only accessible to Users who have the *User Administrator* or the *Admin* permission.

The administrator can review the activated licenses and additional licenses for offline users, high volume scanners, mobile devices, as well as server-side Classification and Extractions volume licenses.



The screenshot shows a dialog box titled 'Licensing Administration'. At the top, there are three tabs: 'Licensing Information' (selected), 'Mobile Licensed Devices', and 'Banned Devices'. The main area is a table with the following data:

License Name	Type of License	Number of Licenses	Available Licenses	Renewal Date	Priority	Expiration Date	Status
System							Active
Concurrent Users	Concurrent Connectio...						Active
Offline Users	Concurrent Connectio...						Active
High Volume Scanners	Concurrent Connectio...						Active
Mobile Devices	Concurrent Connectio...						Active
Classification Pages	Pages						Active
Extraction Pages	Pages						Active

At the bottom right of the dialog box is a 'Close' button.

Figure 165. Licensing Information: List of Activated Licenses

In the additional tabs, the administrator can review the allocated licenses for mobile devices, revoke mobile licenses and move devices to the banned device list.

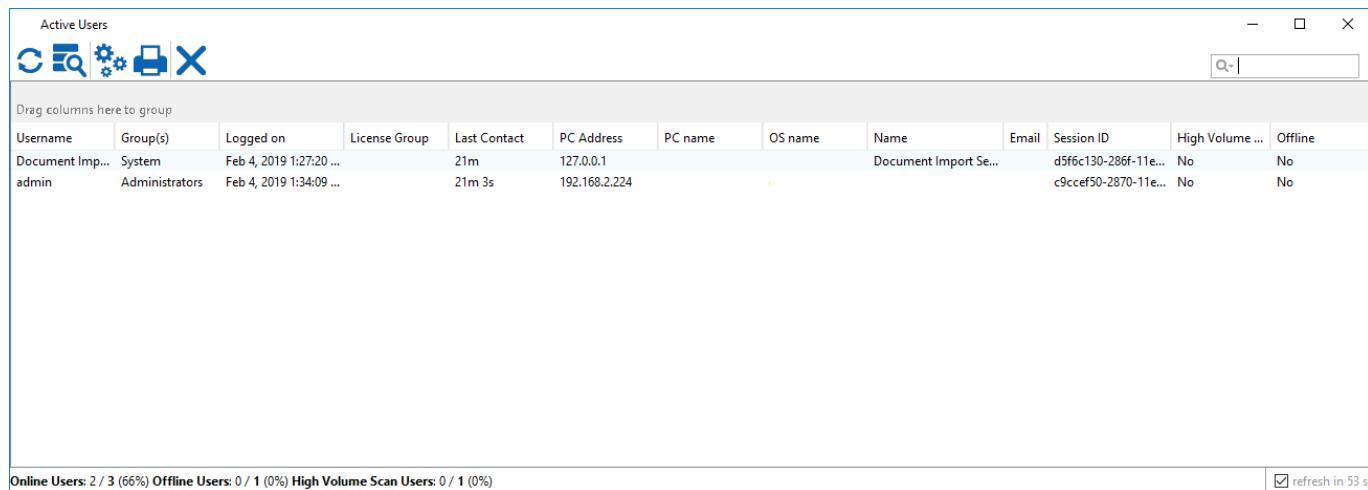
8.4. Active Users

You can see a list of all *Active Users* of the system from the Tools & Options menu → *Active Users...* item. The *Active Users dialog* is available to administrators (*Admin permission*) and users with 'User Administrator' function. The appearance and toolbar of the *Active Users dialog* is identical to the *Batch Manager*

dialog. The list of active users is auto-refreshed according to the refresh rate in the bottom-right value.

You can see the number of *currently logged-in users* and the *maximum number of allowed users* on the bottom left area of the *Active Users* window.

The *Last Contact* column displays the number of seconds between the last time the specific *Client* contacted the *Core Service*. By default, each *Client* sends a heartbeat to the *Core Service* every **10** seconds. Notice though that since the *Active Users* refreshes its data every **60** seconds, this column will display larger values if it is not refreshed before.



Username	Group(s)	Logged on	License Group	Last Contact	PC Address	PC name	OS name	Name	Email	Session ID	High Volume ...	Offline
Document Imp...	System	Feb 4, 2019 1:27:20 ...		21m	127.0.0.1			Document Import Se...		d5f6c130-286f-11e...	No	No
admin	Administrators	Feb 4, 2019 1:34:09 ...		21m 3s	192.168.2.224					c9ccf50-2870-11e...	No	No

Online Users: 2 / 3 (66%) Offline Users: 0 / 1 (0%) High Volume Scan Users: 0 / 1 (0%) refresh in 53 s

8.5. Offline Scanning

Info Input Solution enables users to work offline (disconnected from the *Core Service*) for an extended period, providing the largest possible subset of functions normally provided. Users can create new batches, scan documents and perform indexing while being offline.

Users work offline in offline sessions, which are created when the users log into the system, along with their normal online sessions. Offline sessions have a validity period, which defines how long the user is permitted to work offline. As long as the user is online, this validity period does not come into play. In other words, as long as the user remains online, the offline session is also valid. When the user disconnects however (e.g. leave office, close the VPN connection, etc.), the offline session's validity period is translated to a specific expiration timestamp, which defines the time until which users can create new batches / scan documents. After the offline session expires, the user can only save (locally) the current batch (if any). Afterwards, it is only possible to reconnect to the *Core Service* to upload the locally saved batches.

Apart from the offline session validity period, Info Input Solution Offline Support also defines a maximum for the number of offline sessions that can exist at a time. When the maximum is reached, no new offline sessions can be created: users eligible to work offline that log into the system will not be able to

work offline and when they disconnect from the *Core Service*, the *Thick Client* will simply work in disconnected mode (much shorter functioning period, no local saving and reopening of batches).

8.5.1. Offline session Use Case

The user's offline session starts when the user logs into the system, along with their normal online session. As long as the user remains online, the full set of functions is available. However, when the user disconnects from the *Core Service*, the *Thick Client* switches to offline mode:

- The offline session's expiration timestamp is calculated.
- All server-dependent actions become inaccessible (administrative functions, the Batch Manager, the User Activity monitor, management of Scan Profiles, changing the user password, editing the global script, and opening batches existing in the system).

During the offline session, the user can create new batches / scan documents, save them locally and reopen them at a later time to do further work on them, using the "Open batch" function. Indexing is also available, although any server-side functionality is of course not available. It is possible however to define a Database Action using a client-side data source (e.g. a MS Access database file on the user's computer) and select the option for it to be available when working offline, thus providing look-up and validation to offline users. The following lists the server-dependent functions that are not available when working offline:

- Database Actions (except those that have been explicitly specified to work offline): no validation and / or look-up is performed. If the action validates the value of a field, the field is marked as invalid.
- Counter expressions: they simply evaluate to 0 (zero)
- Remote Procedure Calls on the *Core Service* by scripts.

A new JavaScript object has become available, so that client-side scripts can acquire information related to offline sessions. The object is called *OfflineSupport* and it provides three functions:

- *isInOfflineSession*: boolean indicating whether an offline session has been started, or not. The *Thick Client* may still be connected to the *Core Service* (online mode).
- *isOffline*: boolean indicating whether the *Thick Client* is in offline mode, or is connected to the *Core Service*.
- *getOfflineSessionExpiration*: number (64-bit integer) indicating the offline session expiration timestamp. This function only returns a meaningful value when an offline session has been started and the *Thick Client* is in offline mode.

During the offline session, the user may need to close (exit) the *Thick Client*. Without a connection to the *Core Service*, the user cannot launch the *Thick Client* in the normal way (i.e. directing a browser to a URL), therefore an alternative method is provided: when the *Thick Client* first starts up with offline support enabled, it creates a shortcut on the user's desktop, which the user can use to launch the *Thick Client* in

offline mode. This shortcut points to the *Thick Client* binaries that are cached on the user's computer. The *Thick Client* resumes the offline session and the user can continue working.

At any time during the offline session, the user may reconnect to the *Core Service* and become online again. When this happens, the following also happen:

- The *Core Service* verifies the offline session. The offline session must exist (not have been deleted) in the system for any locally-saved batches to be uploaded.
- The offline session is renewed
- The *Thick Client* starts uploading the locally saved batches
- All server-dependent functions are enabled
- All locally-cached configuration data (Job definitions) is refreshed

At any time, the user may disconnect from the *Core Service* and become offline again. It is not necessary to upload all locally-saved batches to do so. Again the offline session's validity period will be translated to a new expiration timestamp and the user can continue working offline. If the user disconnects while a batch is being uploaded to the *Core Service*, this batch will not be available for opening in offline mode. Its upload will however resume once the user becomes online again.

8.5.2. Offline Administration

Offline-enabled Info Input Solution systems, provide an additional Group permission "Work Offline". The system administrator grants this permission to a group, which it then associates with the users that should be able to work offline. There is no limitation to the number of groups and users that have the offline permission, but at no time can there exist more offline sessions as the system maximum specifies.

The system administrator can view the offline sessions existing in the system in the "User activity" window. Offline sessions display "Yes" in the Offline column in the table. The administrator can delete offline sessions, to lower their number, but then the user working in that offline session will not be able to upload their locally-saved batches when they reconnect. Moreover, their locally-saved batches will be permanently deleted when they become online.

To delete an offline session, it is necessary that the user is not online. As long as a user has the offline permission and the user is online, the system makes sure there is also an offline session for the user, since the user may disconnect at any time. Once the user disconnects and the online session is discarded, the administrator can delete the offline session.

8.6. Announcements

8.6.1. General

Info Input Solution supports the display of *Announcements* to one or more users / user groups, when they log in to the system.

An *Announcement* consists of a *subject*, which is usually one line of text, and a *body* which consist of arbitrary rich-formatted text. An *Announcement* has a *Start* and *End date* and is addressed (e.g. appears) to users of one or more *User Groups*. Announcements are presented to users when they login to the system. Each announcement can be set to appear either:

- each time the user logs in.
- once every day (even if the user logs in more than one times).
- only once.

When users log in, they are presented with a window with a list of all announcements that are addressed to them, according to the above criteria. Users need to acknowledge (e.g. confirm they read) each one of those by clicking on an *Acknowledge* button on the *Announcement Display* dialog, before they are allowed to continue working with Info Input Solution.

8.6.2. Creating - Editing Announcements

The permission *Create global announcement* or *Create group announcement* is required, in order to be able to create and/or edit announcements. To access the *Announcements Administration dialog* click on the *Tools & Options menu* → *Announcements administration...* item:

ID	Subject	Start	End	Created by	Create time	Display Type	Target groups
5	System Maintenance	1/1/20 12:00 AM	1/1/20 9:00 AM	admin	2/4/19 2:57 PM	At every login	global; Administrators; Un...

Figure 166. Authentication Mechanism

You can use the *Announcements Administration* dialog to create and edit announcements using the buttons on the right.

The *Create/Edit Announcement* dialog is used to edit the actual content of the announcement:

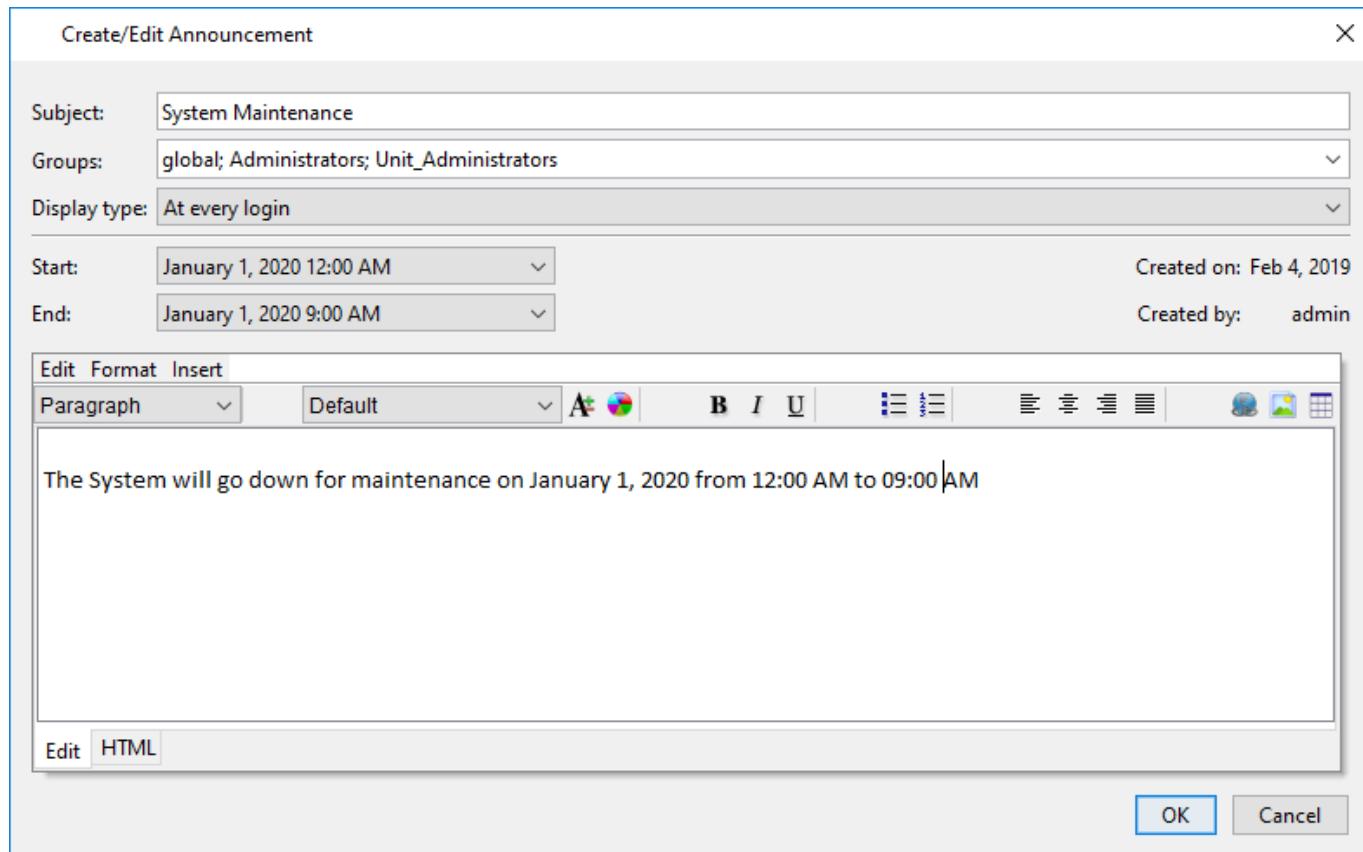


Figure 167. Create/Edit Announcement dialog

Notice the following:

- The *Subject* should be short and descriptive, not more than a single line of text (much like the subject of an email)
- The *Groups* list refers to the groups of users that this announcement is addressed. Notice that:
 - Announcements are always addressed to one or more *User Groups* (never to a single user).
 - Each announcement can be addressed to one or more groups. If a user belongs to more than one group that are addressed by the same announcement, the user only sees this announcement once.
 - A user with the permission *Create group announcement* can create announcements and address them to any of the groups that the user is a member of. A user with the permission *Create global announcement* can create announcements and address them to any group of the system.
 - A user can edit an announcement even if it was created by a different user, if that user has the *Create group announcement* permission and the announcement is addressed to at least one of

the groups the user is a member of. In that case, the user can only edit the groups that s/he is a member of but not the groups that another user may have decided to address this announcement to.

- An announcement always has a *Start date*. This date can either be in the future or in the past. If in the past, the announcement gets in effect immediately after it is saved.
- An announcement may, or may not, have an *End date*. If the announcement has an end date, then it will only appear for the time between the start and end date. If there is no end date, then the announcement will appear indefinitely until someone cancels/deletes it.
- The white space at the bottom of the dialog is used for the actual content of the announcement. Full rich text support is provided, including tables, links and images. The announcement is saved internally as HTML, which you can preview at any time by switching to the *HTML* tab at the bottom. In case you need to write content with complex formatting that this editor may not support, you may use an external HTML editor and paste directly the produced HTML code by clicking on the *HTML* tab at the bottom and pasting the raw HTML code. The HTML renderer used to display announcements does not have full CSS support so complex html may not appear as expected: switching on the *Edit* tab will provide a preview of the announcement as it will appear to the user.

8.6.3. Deleting an Announcement

There are two ways to stop an announcement from being displayed to users (something that will happen anyway once the *End date* passes): You can either choose to completely delete the announcement by clicking on the *Delete...* button from the *Announcements Administration* dialog, or you can edit the announcement itself and set an *End date* that is before the current date.

8.6.4. Display of announcements to users

This is a sample of two announcements as displayed to a user immediately after the log-in step:

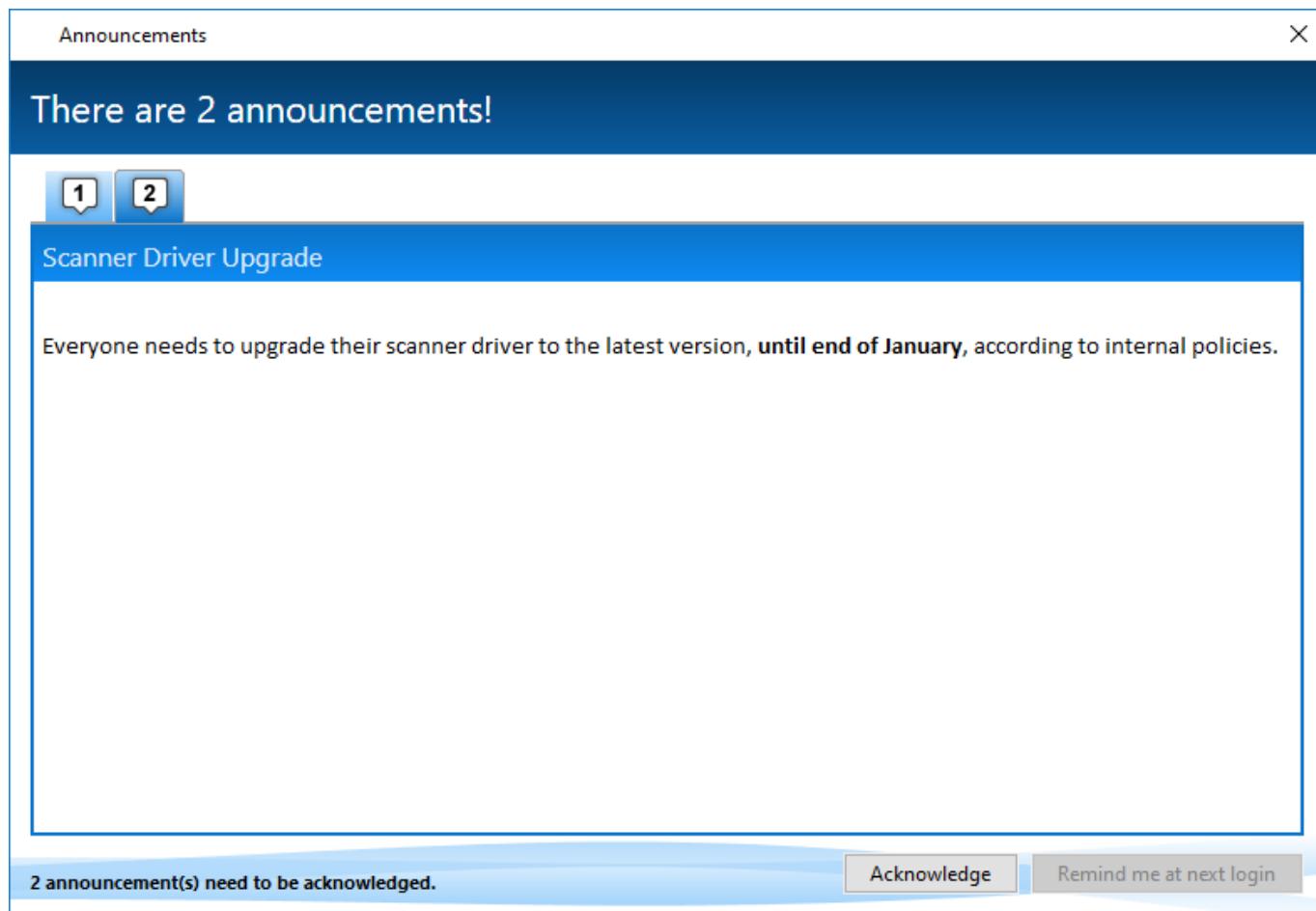


Figure 168. Announcements Display dialog

This dialog will display to the user all the announcements that s/he needs to acknowledge before continuing to use Info Input Solution. The user will need to click on the *Acknowledge* button for each of the announcements (the above dialog displays two announcements in two different tabs). The button *Remind me at next login* appears only for those announcements that are set to appear only *once a day* or *only once*, but the user may choose to see this announcement once more, the next time s/he logs in.

The system is designed so the user needs to click on the *Acknowledge* button for each Announcement: if the user clicks on the *X* button of the window, s/he will see a message saying *You need to acknowledge all announcements before you close this dialog.*

8.6.5. Viewing announcements after initial log-in

Users can view all active announcements at any time but clicking on the *Tools & Options menu* → *Announcements...* item. The *Announcements Display* dialog appears but without the *Acknowledge* button at the bottom; the user can read again any announcements and close the dialog when done.

Notice that this function always displays the announcements that the user viewed when s/he logged in

and does not retrieve from the *Core Service* any new announcements that may have been published after the user logged in. If the user needs to check whether there are any new announcements, s/he will need to re-login (this is by design since announcements are meant to be displayed only during login, and this viewing is used only as a reminder facility).

9. Task Filters

The customization of Task Filters provides the Administrator with the ability to customize the Batch Menu by adding shortcuts that will open Tasks from a specific filtered queue. The section below explains how this customization is achieved and gives an explanation of the configuration GUI.

9.1. Configuring and customizing Task Filters

The goal of the Task Filters Customization is that the users will be able to quickly open a Batch/Task that is in a custom queue. For every custom queue, a shortcut/link is created in the Batch menu.

To configure and customize any *Task Filter* based on your needs, select the Task Filters customization option from within the Tools & Options Menu (main menu) in the Thick Client.

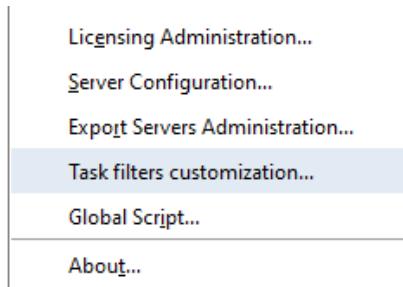


Figure 169. "Task Filters Customization" Menu Item

This will open the following dialog:

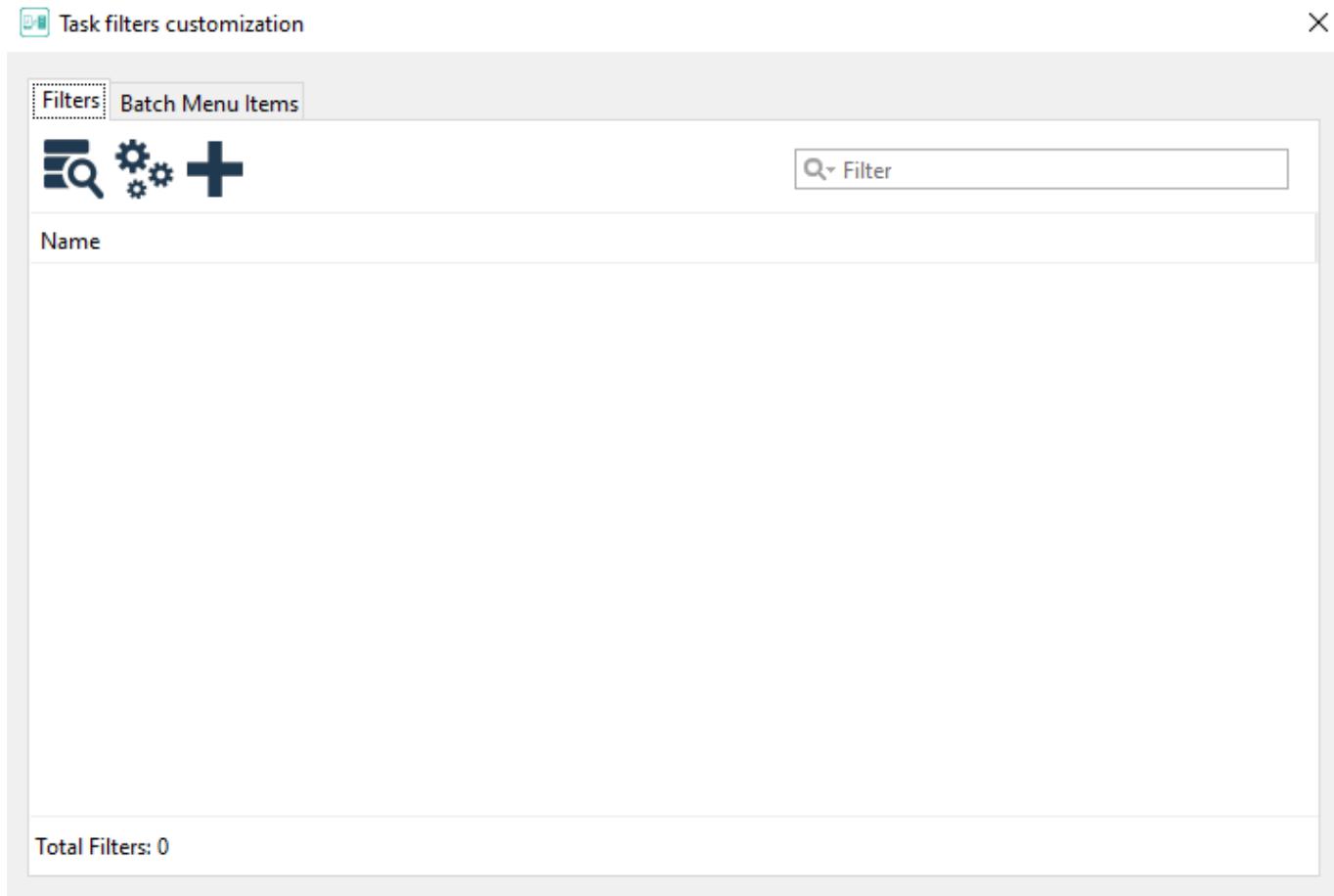


Figure 170. "Task Filters Customization" configuration dialog

This dialog contains two tabs, the Filters and the Batch Menu Items tab.

Here is an explanation of the items in this dialog, when the Filters tab is selected:

Search (Magnifying glass icon)

This can be used to enable searching among the available Task Filters. A search bar will be displayed at the bottom of this dialog upon this icon is selected

Create new Task Filter (Plus icon)

The plus icon can be selected to create a new Task Filter

Filter text box

This text box can be used to dynamically narrow down the items displayed in the Task Filters table

Name column

This is the column that displays the name of each Task Filter

Table options

Additional settings can be applied to the table containing the available Task Filters

To create a new Task Filter, select the "Create new Task Filter" button from the "Filters" tab. Afterward, the following dialog will be displayed:

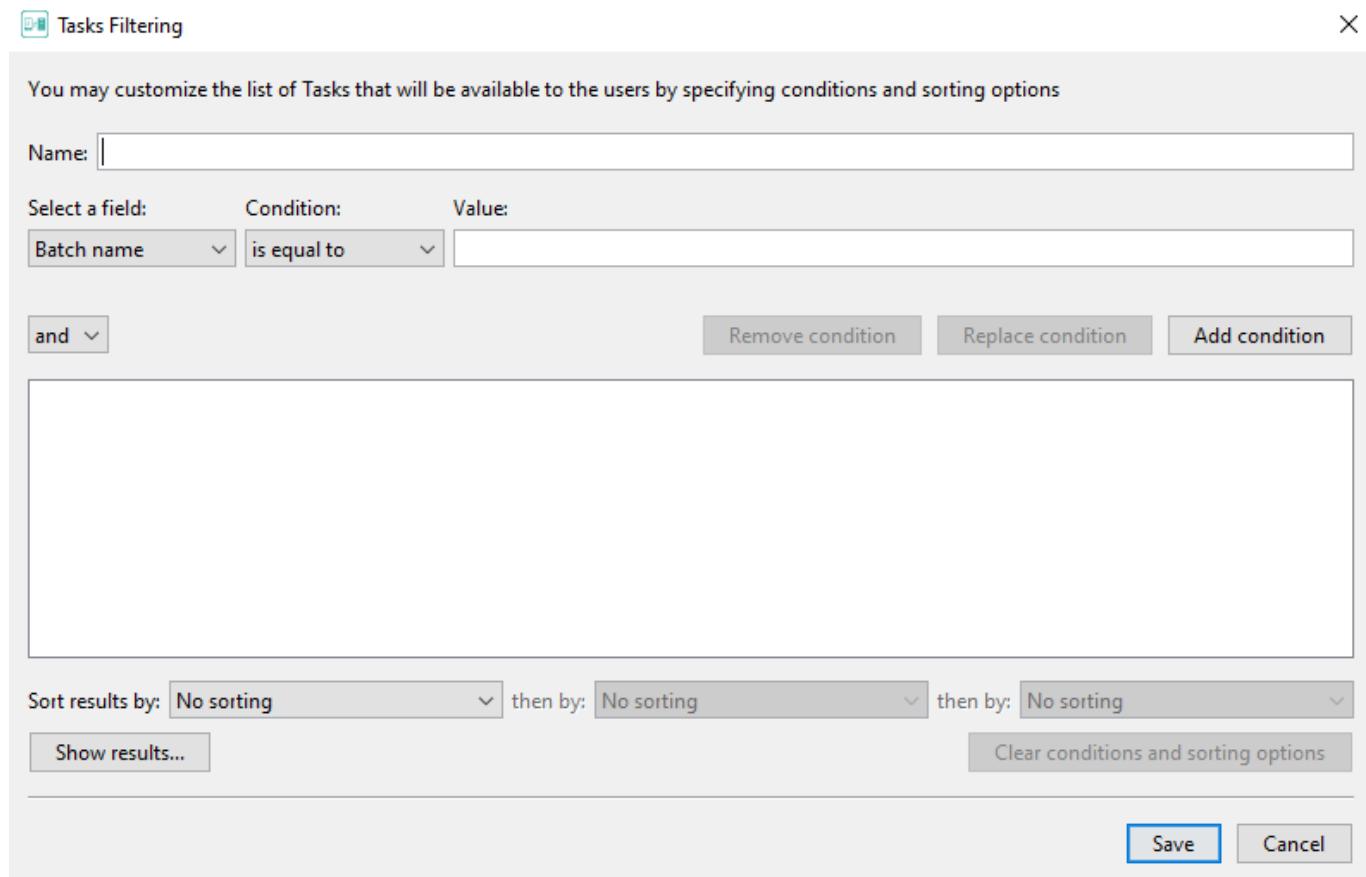


Figure 171. Creating a new Task Filter ("Filters" tab)

Here is an explanation of the sections in the Tasks Filtering dialog shown above:

- In the top section, the conditions of the Task Filter can be defined. By selecting the "Add condition" button, the conditions defined are set and added in the table located in the center
- The center section shows the Task Filter conditions that are currently in place
- The bottom section can be used to customize the sorting. The "Show results" button can be used to show the results after this Task Filter is used

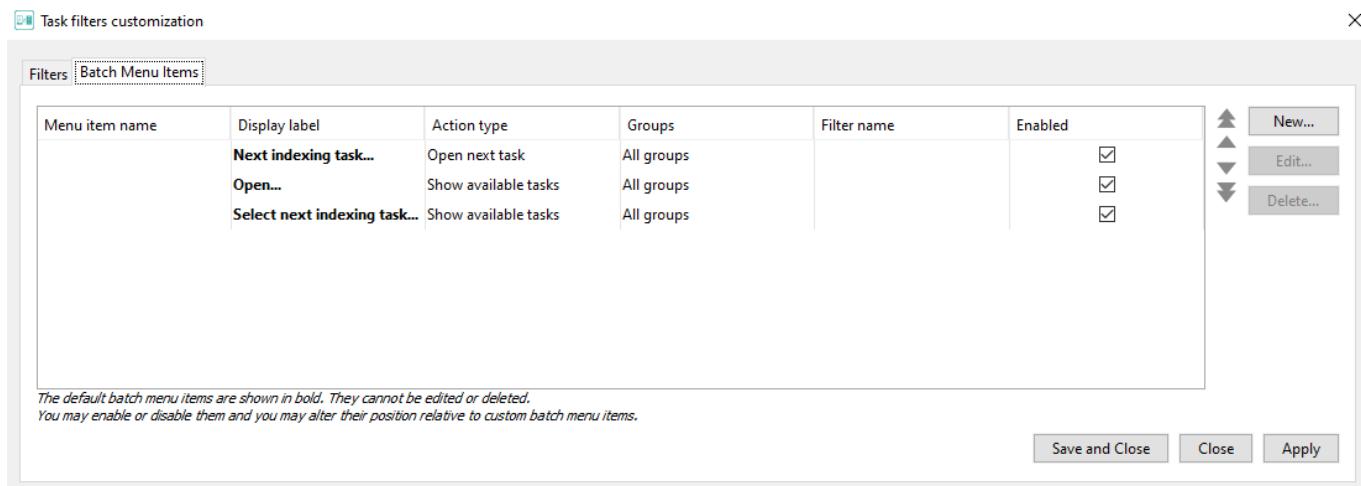


Figure 172. "Batch Menu Items" tab

Here is an explanation of the Batch Menu Items tab:

- Menu Item name: This is the name of each menu item
- Display Label: This is the item name that will be displayed in the Batch menu
- Action Type: This is the action that will be executed when selecting the specific item. This can be either "Open next available task", or "Show all available tasks and allow user to select"
- Groups: These are the User Groups in which this menu item will be available
- Filter Name: The name of the Filter that will be used. The filters are defined in the first tab
- Enabled: Whether the menu item will be enabled or not
- New...: Selecting this will add a new Batch menu item definition
- Edit...: Selecting this will edit the selected Batch menu item definition
- Delete...: Selecting this will delete the selected Batch menu item definition

To add a new Batch Menu Item select the New button. Afterward, the following dialog will be displayed:

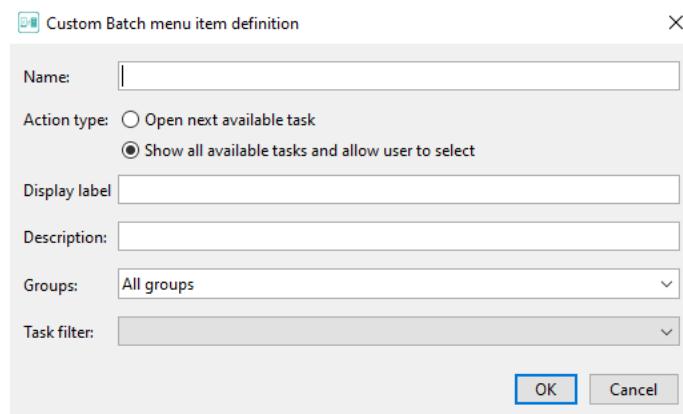


Figure 173. Custom Batch menu item definition

Here is an explanation of the Custom Batch menu item definition dialog shown above:

- Name: The name of the Batch Menu Item
- Display label: This is the display name of the Batch Menu Item that will be displayed in the Batch Menu
- Description: This is a description of this particular Batch Menu Item that will be also displayed in the Batch Menu
- Groups: The User Groups for which this Batch menu item will be available
- Task Filter: The Task Filter that will be used for this Batch Menu Item. (The filters are defined in the Filters tab, as mentioned above)

After having this configuration in place the Task Filters configured will show up in as new Batch menu items in the Batch Menu, in both Thick Client and HTML Client. See the images below:

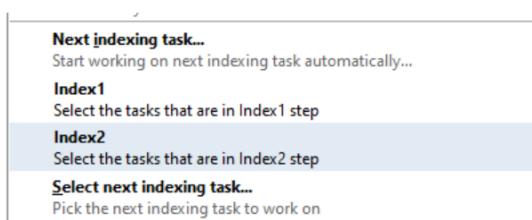


Figure 174. Thick Client Batch Menu

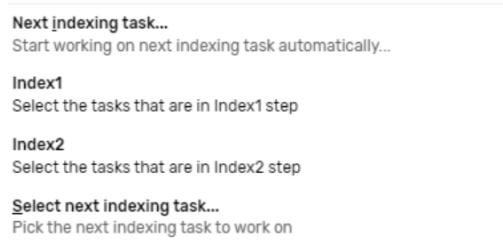


Figure 175. HTML Client Batch Menu

The Menu items Index1 and Index2 are customized Task Filters. When selecting any of the Task Filters you will be able to select the next Task from the customized queue dialog that will be displayed, based on your needs.

10. Custom Fields

The Custom Fields are customizable Fields that can be set per Node and are available system wide.

These fields are available in all the Open and Get Next Indexing Task windows, custom or default, as well as in the Batch Manager window.

Additionally, they are available to be used as conditions in every Task Filter configuration, or even at the Batch Filter that can be implemented at the Server Side.

10.1. Custom Fields Instructions

To configure the Custom Fields, it is necessary to add them in the Server Configuration parameters `scanapp.client.node_custom_field_labels` and `scanapp.client.node_custom_fields`.

Both these parameters accept a comma separated list of values. For every field, both properties need to be configured with the name and the type of the Field accordingly.

The property `scanapp.client.node_custom_field_labels` accepts the following values:

For String values:

- `customString1`
- `customString2`
- `customString3`

For Integer values:

- `customInt1`
- `customInt2`
- `customInt3`

For Date values:

- `customDate1`
- `customDate2`
- `customDate3`

For Decimal values:

- `customDecimal1`
- `customDecimal2`

- customDecimal3

The property `scanapp.client.node_custom_field_labels` accepts a comma separated list with strings that are expected to be the labels of the Fields.

Here is a sample configuration:

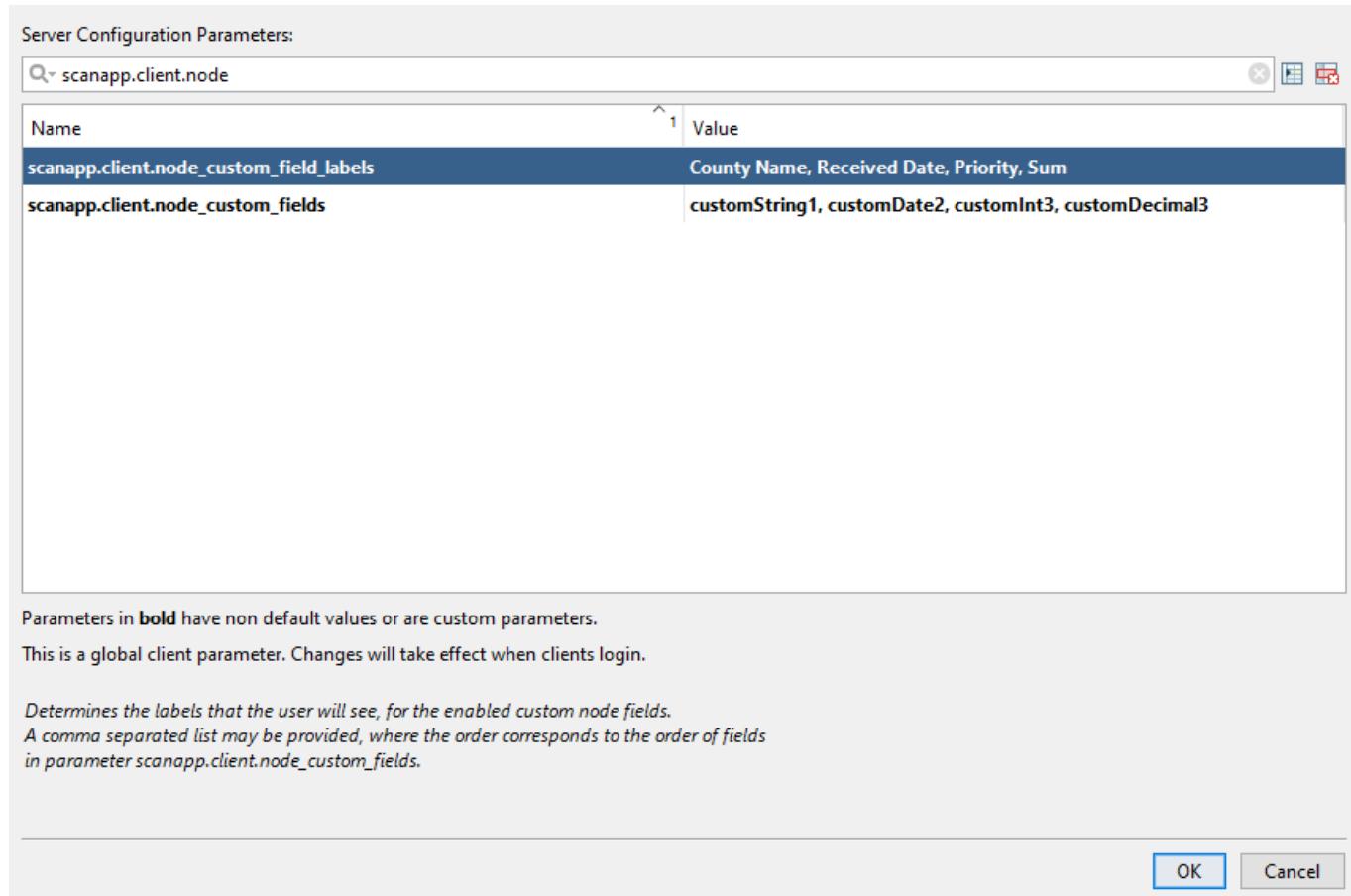


Figure 176. Server configuration tab

`scanapp.client.node_custom_field_labels` and `scanapp.client.node_custom_fields` Server Configuration Parameters

This configuration will create four additional columns that will be available system wide. For example, see the Batch Manager dialog in the screenshot below:

Batch Manager

Received on	Upload state	Priority	Sum	County Name	Received Date	New Batch
8	234.43		NewYork	Mar 31, 2023, 3:00:00 AM	No	
2			Texas	Mar 2, 2023, 2:00:00 AM	No	
1			Michigan	Mar 1, 2023, 2:00:00 AM	No	

Total items: 3, last updated: 9:40:44 PM refresh in 49 s auto-refresh every Secs.

Figure 177. Batch Manager Custom Task Fields outcome

The values of these Fields can only be set programmatically, when using the following Node functions:

```
//For string fields
node.setCustomString1(<value>), node.setCustomString2(<value>),
node.setCustomString3(<value>)

//For date fields
node.setCustomDate1(<value>), node.setCustomDate2(<value>), node.setCustomDate3(<value>)

//For integer fields
node.setCustomInt1(<value>), node.setCustomInt2(<value>), node.setCustomInt3(<value>)

//For Decimal fields
node.setCustomDecimal1(<value>), node.setCustomDecimal2(<value>),
node.setCustomDecimal3(<value>)
```

To get the values of these Fields programmatically the following Node functions can be used:

```
//For string fields
node.getCustomString1(), node.getCustomString2(), node.getCustomString3()
//For date fields
node.getCustomDate1(), node.getCustomDate2(), node.getCustomDate3()
//For integer fields
node.getCustomInt1(), node.getCustomInt2(), node.getCustomInt3()
//For Decimal fields
node.getCustomDecimal1(), node.getCustomDecimal2(), node.getCustomDecimal3()
```

It is very important to always set the values using the correct data types. For example the date custom Fields can only accept Date objects.

The example below demonstrates how to set a value to a Batch level CustomDate2 Field, when the value is taken from an Index Field:

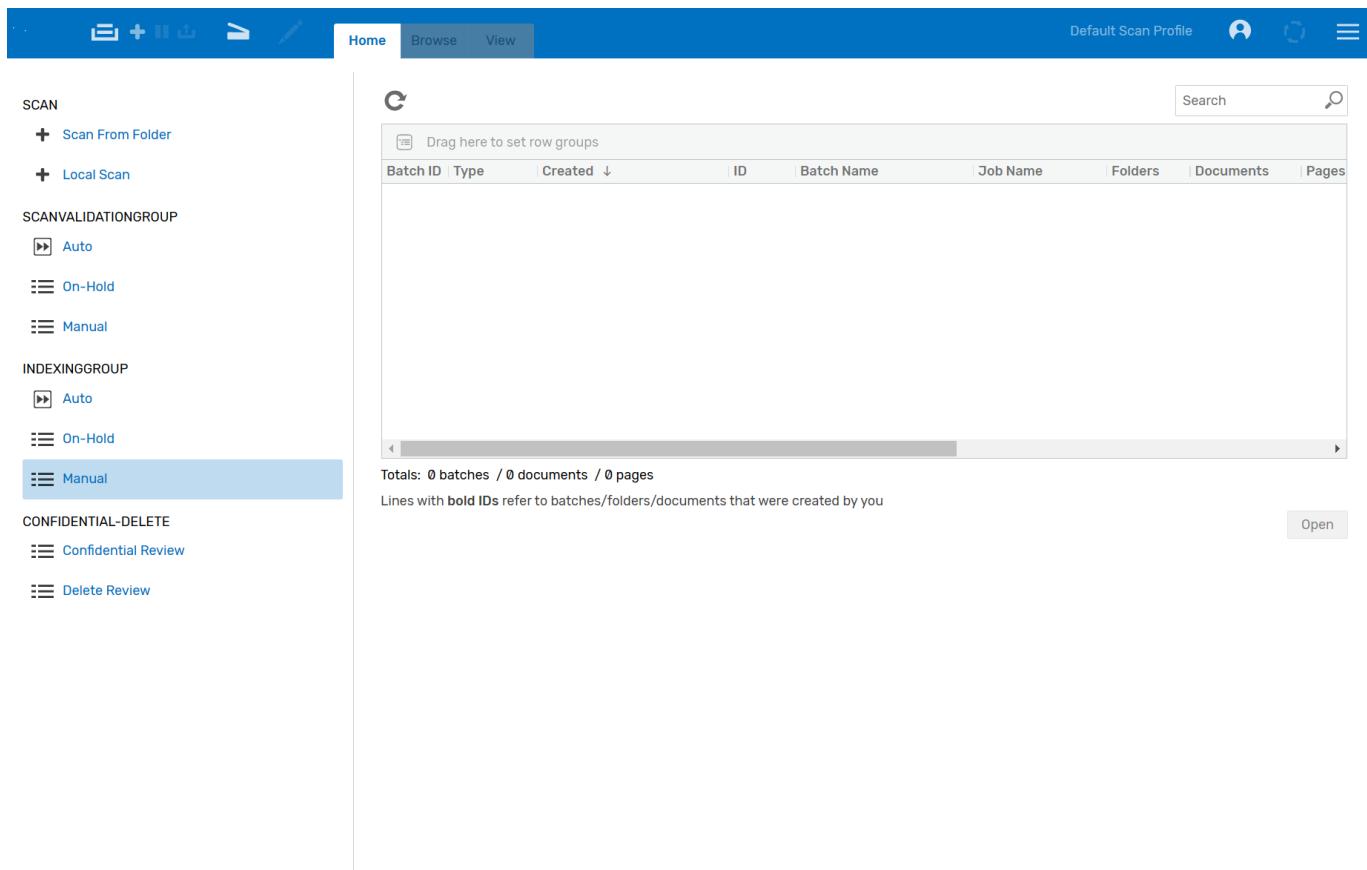
```
var date = new Date(batch.fields["Date"].value);
batch.setCustomDate2(date);
```

11. Home Tab

This is a feature available only in the HTML Client.

The custom Home Tab configuration provides a flexible way to add multiple items in the Home Screen of the HTML Client that will display different queues or perform different actions.

The Home Tab will always be selected when no Batch is opened in the HTML Client, and disabled when a Batch is opened. See the example image below:



Batch ID	Type	Created	ID	Batch Name	Job Name	Folders	Documents	Pages

Totals: 0 batches / 0 documents / 0 pages
Lines with **bold** IDs refer to batches/folders/documents that were created by you

Figure 178. Home Tab

The Home Tab consists of two panes:

- The left pane, that contains a vertical list of links, each one associated with a specific link-type. These link items are Home Menu Items.
- The main pane hosts the main content of the Home Tab that changes according to the link-type that is invoked when the user selects a link from the left pane.

11.1. Instructions

To enable the custom Home Tab, the `scanapp.client.html.homeview_enabled` Server Configuration Parameter should be set to true. To do so, click on the Tools & Options Menu (Main Menu) > Server Configuration option using the Thick Client.

This will just enable the Tab that will remain focused only when a Batch is not opened at the HTML Client. To customize the Items that will be shown in the left panel, and the functionality that will take place, the Global Script functions `getHomeMenuItems` and `getGridCustomizations` must be customized accordingly.

11.1.1. The `getHomeMenuItems` function

If this function returns false, the Home Tab sidebar does not get updated or cleared. This is useful for preventing re-rendering the entire sidebar in case of a static list of links. The items that are returned should match the items that are defined in the Task Filters. Refer to the Task Filters section for more details on how to customize the Task Filter Items.

The display order of the groups and links depends on the order of the returned array (groups have priority).

The return value of the function is an object with the following structure:

```
{  
  items: HomeMenuItem[],  
  defaultItem: HomeMenuItem,  
  showGroups: boolean,  
  collapsible: boolean  
}
```

The items in this JSON are the following:

items

An array of Home Item item objects; each one corresponds to a link that is displayed.

defaultItem

A Home Tab item of type display-tasks that will be the default one when switching to the Home Tab.

showGroups

Whether groups will be visible or not.

collapsible

Whether groups will be collapsible or not.

The Home Tab item type should be an object that will has the following format:

```
{  
  name: string  
  label: string  
  type: "display-tasks" | "load-single-task" | "show-new-batch-dialog" | "new-batch"  
  enabled?: boolean  
  groupName?: string  
  filterName?: string  
  newbatchCallback?: (itemName: string, jobs: IJob[], scanProfiles: IScanProfiles[]) =>  
  NewBatchReturnValue  
}
```

The items in this JSON are the following:

name

This is a unique name for the item; will be used as an internal identifier and it will not be displayed

label

This will be the actual label displayed in the Home Tab link

type

The type of the action when the link is selected. There are 4 available types:

display-tasks

displays a list of filtered tasks for a user to select; the filterName property must be set

load-single-task

Opens the next available task using the given Task Item; the filterName property must be set in the filters from the Task Items configuration mentioned above

show-new-batch-dialog

Opens the New Batch Dialog for the user to create a new batch

new-batch

creates a new batch; the **newbatchCallback** callback is used for customizing the creation of the New

Batch dialog.

enabled

Whether the item is enabled or not. `groupName`: the group in which the item belongs to; If undefined, the item will appear at the top without any grouping

filterName

The name of the filter that is defined in the Task Filters Customization

newbatchCallback

This is required if the item is of type new-batch, and it should be set within the function that will be called when the user selects the item. The type of the callback is the following: `(itemName: string, jobs: IJob[], scanProfiles: IScanProfiles[]) => NewBatchReturnValue`

itemName

The name of the Home Menu item that was selected.

jobs, scanProfiles, return-value

Refer to the `newBatch()` function of the Global Script.

11.1.2. The `getGridCustomizations` function

This function is called each time a Task List is displayed. It can be used to customize the columns display, their order, the row sorting and row grouping. This function will also be called when the Open item or the Select next index task is selected from the Batch Menu.

- The name argument is the name of the Home Menu Item of type display-task that was selected, or it can be open-task-to-scan when the Open item from the Batch Menu is selected or it can be open-task-to-index when the Select Next Task to index is selected.
- The allcolumns argument is an array of strings that represent the columns of the list to be displayed.

The following JSON Object should be returned:

```
return {  
  columns: string[],  
  sortby: {column: string, direction: "asc" | "desc"}[],  
  groupby: string[]  
}
```

The items in this JSON are the following:

columns

This property is an array of column names that are to be displayed. The columns ordering will be the same as the ordering in the array.

sortby

This property is an array of JSON objects that have 2 properties each: the column which is the name of the column to sort by and direction which is either asc for ascending or desc for descending direction. The order of multiple sorted columns will be the same as they appear in the array.* groupby: This is the property of the array names that the list will be grouped by. The grouping order will be the same as they appear in the array. Any column name that is not matched by an available column will be ignored.

The list of the available column names, to be used in the getGridCustomizations function can be found below:

```
'id', 'nodeType', 'batchId', 'batchName', 'jobName', 'numFolders', 'numDocs',  
'numPages', 'createUsername', 'created', 'priority', 'currentStepDisplayName', 'privat',  
'description', 'status', 'customString1', 'customString2', 'customString3',  
'customInt1', 'customInt2', 'customInt3', 'customDecimal1', 'customDecimal2',  
'customDecimal3', 'customDate1', 'customDate2', 'customDate3'
```

11.2. Custom Home Tab Examples

Example of a Home Menu item that belongs to a Group named Scan and creates and scans a new Batch using the Job named Technical Description:

```
var item1 = {  
    name: "scan-technical-description",  
    enabled: true,  
    groupName: "Scan",  
    label: "Technical Description",  
    type: "new-batch",  
    newbatchCallback: (itemName, jobs, scanProfiles) => {  
        return {  
            actionName : 'CREATE',
```

```
    jobName : 'Technical Description',  
  }  
}  
}
```

Example of creating 4 Home Menu links, with the second one being the default and having the groups displayed and not being collapsible. This example requires that the following are already defined:

- a Job Setup, named Contracts
- a Task Filter, named filter-contacts
- a Task Filter of type Open next task, named Next Contract

```
var item1 = {  
  name: "create-batch",  
  enabled: true,  
  label: "Create a new Batch",  
  type: "show-new-batch-dialog"  
};  
  
var item2 = {  
  name: "list-contacts",  
  enabled: true,  
  groupName: "Show all",  
  label: "Contracts",  
  type: "display-tasks",  
  filterName: "filter-contacts"  
};  
  
var item3 = {  
  name: "getnext-contract",  
  enabled: true,  
  groupName: "Work on the next pending",  
  label: "Contract",  
  type: "load-single-task",  
  filterName: "Next Contract"  
};
```

```
var item4 = {
  name: "scan-contract",
  enabled: true,
  groupName: "Scan",
  label: "Contract",
  type: "new-batch",
  newbatchCallback: function(itemName, jobs, scanProfiles) {
    return {
      actionName : 'SCAN',
      jobName : 'Contracts',
      batchPrivate : false,
      batchPriority : 10,
      batchProperties : {
        prop1 : 'value1',
        prop2 : 'value2'
      }
    }
  }
};

return {
  items: [item1, item2, item3, item4],
  defaultItem: item2,
  showGroups: true,
  collapsible: false
}
```

Example of how to display only the columns batchId, nodeType, and created, sorted by the created column in descending order:

```
return {
  columns: ["batchId", "nodeType", "created"],
  sortby: [{column: "created", direction: "desc"}]
}
```

12. Server Administration

The Info Input Solution administrator can manage the database configuration or create a new one, review and import licenses, check and restart the application component services and configure various *Core Service* parameters.

12.1. Administration Utility

The Administration Utility can be found in *Start menu* → *Info Input Solution* folder. At the first tab, the system administrator can view the current setup of the active Database connection, change between different Database configurations or create a new Database configuration for *Info Input Solution*. An installation can have multiple database configurations, but only one can be active.

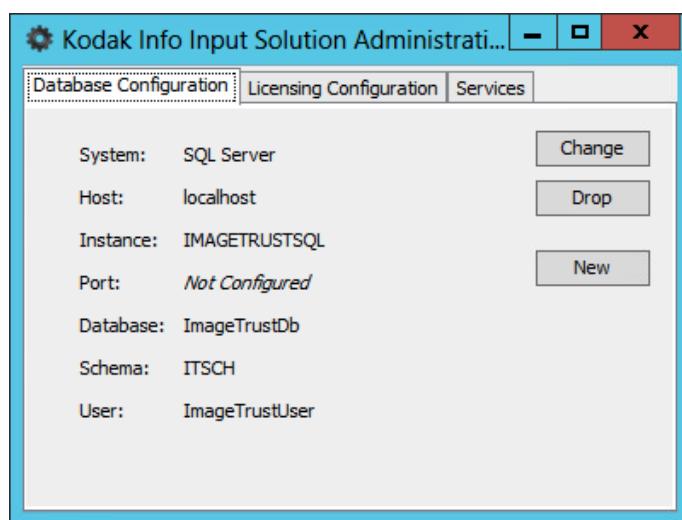


Figure 179. Administration Utility - Database Configuration

The second tab can be used for importing License files, as well as for reviewing the properties and expiration dates of the currently installed Licenses.

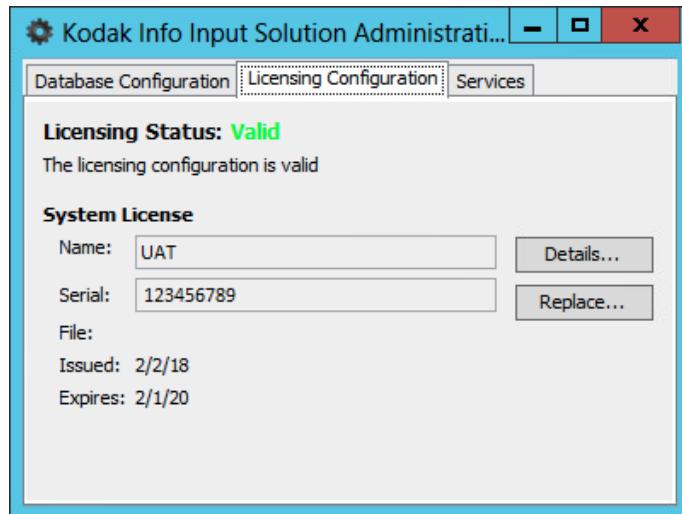


Figure 180. Administration Utility - License Configuration

The third tab displays the installed Local Services, necessary for running the Info Input Solution application. An administrator can review their current status, and also start / restart each one of them.

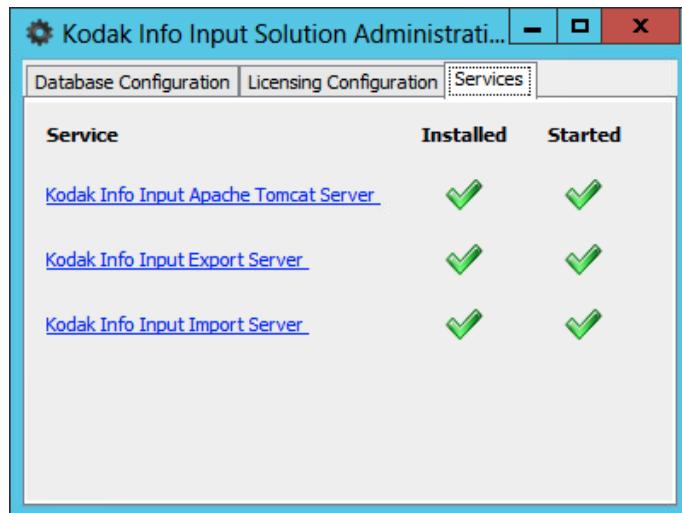


Figure 181. Administration Utility - Services Status

12.2. Server Configuration Parameters

The system administrator has the ability to review and manage a set of *Core Service* configuration parameters, via the *Tools & Options menu* → *Server Administration...* option.

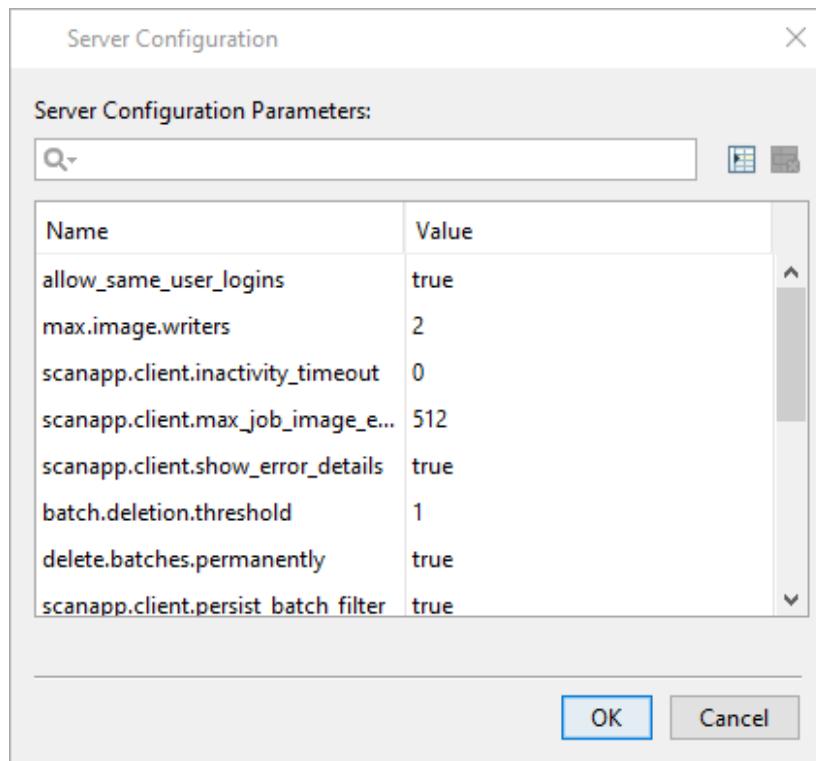


Figure 182. Server configuration parameters

Some of the parameters listed above are global *Client* parameters, and changing the value will immediately apply to all new logins. When modifying a *Core Service* parameter, restarting all *Core Services* is required for the changes to take effect.

12.2.1. Global Client parameters

scanapp.client.node_custom_fields

Determines which custom node fields are enabled so that they can be seen by the users. A comma separated list of the custom field names may be provided, for example "customInt1,customDate1". By default, all custom node fields are disabled.

scanapp.client.node_custom_field_labels

Determines the labels that the user will see, for the enabled custom node fields. A comma separated list may be provided, where the order corresponds to the order of fields in parameter *scanapp.client.node_custom_fields*.

scanapp.client.use_check_scanning

Whether check scanning with SilverBuller Ranger is enabled. The default value is false.

scanapp.client.ranger_config

SilverBullet Ranger configuration

scanapp.client.use_relative_page_coordinates_and_point_size

Use relative page coordinates for Annotations and BarcodeData. Default value is true. This parameter determines the behavior of the public API functions that have to do with coordinates and sizes in Annotations and BarcodeData. By default these functions return relative coordinates, relative width or height and stroke, outline or font size in points. If this parameter is set to false, then they return absolute coordinates, absolute width or height and stroke, outline or font size in pixels.

scanapp.client.batch_manager_max_results

Determines the maximum number of tasks that can be displayed in the Batch Manager dialog. The default value is 1000. You may set it to -1, in order to allow the Batch Manager dialog to display all available tasks. It is recommended, for better performance, that this value is not set higher than 5000.

scanapp.client.indexing_evaluate_field_values

Indexing panel should evaluate assigned field values. Default value is false. This parameter determines whether field values should be treated as expressions and evaluated when loading them on the indexing panel.

scanapp.client.default_page_dpi

This is the DPI image property that will be used for imported image files, or image-based PDF files, when not available.

scanapp.client.max_job_image_edge

Maximum edge size for a Job image in pixels. Default value is 512. This parameter determines the maximum allowed edge (width or height) of an image assigned to a Job *scanapp.client.html.max_concurrent_server_ops* ::This is the maximum number of concurrent RPC requests allowed, e.g. invocations of `dispatch.invokeOnServer()`

scanapp.client.html.homeview_enabled

Whether the Home Tab will be enabled or not.

scanapp.client.show_error_details

Show error details. Default value is true. This parameter determines whether full error details will be shown to the client when an error occurs.

scanapp.client.html.theme_config_accent_color

HTML Color code for the accent color of the HTML Client theme. The color is used to give emphasis to areas/elements that require attention.

scanapp.client.html.batch_manager_columns_lock_order

Whether to lock the order of the columns of the Batch Manager Dialog. Set 'true' to lock, 'false' to

unlock.

scanapp.client.tasks_list_max_results

Determines the maximum number of tasks that can be displayed in the Open task dialog, Select next indexing task dialog and in the Home Display Tasks lists. The default value is 1000. You may set it to -1, in order to allow the lists to display all available tasks. It is recommended, for better performance, that this value is not set higher than 5000.

scanapp.client.html.initial_cct_detect_delay_ms

Number on milliseconds that are allowed to detect whether CCT service is running or not before displaying a blocking window on top of the HTML Client that prompts the user to download it. This parameter only makes sense if RequireCCTOnStartup is true. Default value: 3000

scanapp.client.html.keyboard_shortcuts

Administrator provided keyboard shortcuts for the HTML client

concurrent.users.audit.period.sec

Determines how often (in seconds) the concurrent users are reported and logged to the audit file. The default value is 0, which means that auditing is disabled. The period cannot be less than 60 sec.

scanapp.client.html.rendcache_size_mb

Cache size in MB for images/rendition in HTML Client memory.

scanapp.client.html.scan_from_flatbed_if_adf_is_oop

If the scanner has both ADF and Flatbed, this option will trigger attempting to scan from the Flatbed if the ADF is out of paper on scan start.

scanapp.client.html.use_cct

Whether the HTML Client will use CCT or not. If true, the client will attempt to connect to CCT service on startup and will also provide an indicator to the user about the connection status. If false, then no indication whatsoever is given to the user. The HTML Client works totally independently.

scanapp.client.node_customdate1_format

Determines the display format for a node's customDate1 field.

scanapp.client.node_customdate2_format

Determines the display format for a node's customDate2 field.

scanapp.client.node_customdate3_format

Determines the display format for a node's customDate3 field.

scanapp.client.html.log_to_server_context_id

This is a string that is sent to the server when remote logging is enabled. It is used in the server's log4j setup to distinguish logging arriving from clients. If you leave this empty it defaults to the client internal IP address. In production environment it may make sense to set this to a user-id or another context value that may help in identifying the browser/user/session that is sending the logging information.

scanapp.client.html.theme_config_base_color

HTML Color code for the base color of the HTML Client theme. Most elements will use this color.

scanapp.client.html.custom_logo

The URL of an image to be used as a custom logo. It can be relative or absolute and the formats supported are PNG and SVG.

scanapp.client.html.supports_search

Responsible for toggling the Search toolbar icon and functionality.

scanapp.client.html.prompt_for_cct_download

Whether to prompt the user to download and install CCT if not found during initialization. If RequireCCTOnStartup is true, the user will get offered a download link of CCT on the CCT required prompt. If RequireCCTOnStartup is false and PromptForCCTDownload is true then a notification is being displayed to the user with the download link of CCT. Default value: false. If RequireCCTOnStartup is true but PromptForCCTDownload is false then the application will block if it doesn't detect CCT but it will not offer a link to download it. Use this configuration if you require CCT but want to control when and how the users install CCT in their local machines or when users do not have installation permissions.

scanapp.client.html.batch_manager_groups_collapsed

Whether the Batch Manager will expand or collapsed the configured row groups by default or not.

scanapp.client.html.use_aggrid_enterprise

Use the Licensed Enterprise features of AgGrid for the BatchManager dialog (and elsewhere) or not. This enables the advanced BatchManager table features (Enterprise features of the AgGrid third party Dynamic Grid Component). NOTE: Besides setting this to 'true', the associated License property is also required to be enabled in order for these features to get enabled.

scanapp.client.html.batch_manager_grouped_columns

Configure which columns will be grouped by default in Batch Manager for all clients. Type in each column name separated by comma.(nodetype, nodeid, nodeorder, parentid, batchid, batchname, description, jobid, jobname, foldercount, documentcount, pagecount, creator, creatorgroup, createddate, priority, queue, mode, status, lockedby, lockdate, splitfrombatchid, copiedfrombatchid,

privatebatch, newbatch, uploadstate)

scanapp.client.html.log_to_server

Whether client logging is directed to server or not. Default value: true

scanapp.client.inactivity_timeout

Inactivity timeout in minutes. Default value is 0. This parameter determines after how many minutes of inactivity, the client will be forced to logout. Valid values are 0 to 500.

scanapp.client.html.disable_adf_detection_for_oop

Whether to disable the automatic detection of ADF (Automatic Document Feeder) for the reason of determining whether the out-of-paper handling should be disabled. The out-of-paper handling fails to perform properly with scanners that have both a flatbed and ADF. In order to disable the OOP handling for those scanners we need to detect whether the ADF is really used or not; yet some scanners may fail to report that properly. If this happens, the OOP is permanently disabled, something that is not desirable in most cases. Setting this to true allows the OOP handling to operate as expected.

scanapp.client.html.disable_oop_handling

Whether to disable automatic handling of out-of-paper when scanning. Default value is true. Set it to false to have the scanner stop when it goes out of paper

scanapp.client.node_createdate_format

Determines the display format for a node's creation date.

scanapp.client.html.cctcache_size_mb

Cache size in MB for images in HTML Client memory while scanning; used only for 8-bit images. Set to zero (0) to disable cache.

scanapp.client.persist_batch_filter

Persist user defined Task Filter. Default value is true. This parameter determines whether the user defined Task Filter is persisted on the database, so that it can be reused when the user logs in again.

scanapp.client.html.default_browser_context_menu_in_indexing

Whether the default browser context menu should be allowed in input controls in indexing

scanapp.client.html.batch_manager_show_grouping

Whether the Batch Manager will show the groups header or not by default.

scanapp.client.html.reuse_cct_session

Whether the HTML Client will attempt to reuse an established CCT session or create a new one upon

initialization.

scanapp.client.batch_audit

Enable Batch audit. Default value is false. This parameter determines whether the client will transmit audit information to the server, whenever a Batch is suspended or closed.

scanapp.client.index_tasks_max_results

Determines the maximum number of tasks in the Index queue that can be displayed, when the task selection dialog is opened. The default value is 1000. You may set it to -1, in order to allow the dialog to display all available tasks. It is recommended, for better performance, that this value is not set higher than 5000.

scanapp.client.html.require_cct_on_startup

Whether the CCT service should be detected and running during initialization. If true, then the HTML Client will block until CCT becomes available. If CCT becomes unavailable at a later point (e.g. after startup), the client will continue to work. This flag only sets the initial state. Default value: false

scanapp.client.html.batch_manager_columns_list

Configure which columns will be displayed in the Batch Manager Dialog. Type in each column name separated by comma. (nodetype, nodeid, nodeorder, parentid, batchid, batchname, description, jobid, jobname, foldercount, documentcount, pagecount, creator, creatorgroup, createddate, priority, queue, mode, status, lockedby, lockeddate, splitfrombatchid, copiedfrombatchid, privatebatch, newbatch, uploadstate). Leave it empty to display all columns.

scanapp.client.node_lockeddate_format

Determines the display format for a node's locked on date.

scanapp.client.scan_tasks_max_results

Determines the maximum number of tasks in the Scan queue that can be displayed, when the task selection dialog is opened. The default value is 1000. You may set it to -1, in order to allow the dialog to display all available tasks. It is recommended, for better performance, that this value is not set higher than 5000

scanapp.client.html.cct_url

This is the OpenText/Captiva cloud toolkit runtime re-distributable. Provide a fully qualified url starting with http, otherwise the url will be appended to the HTML Client static files root.

scanapp.client.html.show_stack_traces

Use this if stack traces should be shown in HTML Client error dialogs.

scanapp.client.scanner_imprinting

Set to true to enable scanner imprinting handling. If false the application will not handle imprinting although it might still take place based on the scanner configuration.

scanapp.client.full_size_preview(Thick Client Only)

Set to true to allow the image preview to be larger than 400 pixels when the dialog is enlarged.

- Clarification: The dialog itself will not be automatically enlarged (or maximized) if this Server Configuration parameter is set to true, but the preview of the image that was scanned/imported.

scanapp.client.font (Thick Client Only)

Changes the font of the Thick Client. A Font Family name that is available in all end-user workstations must be used.

scanapp.client.font_size (Thick Client Only)

A valid integer must be used to change the size of the letters.

scanapp.client.max_failed_communication_attempts

Determines after how many failed heartbeat attempts (one every 10 sec), the Thick Client application will change to "Communication Lost" mode and display the following message. In this mode, the Thick Client session is not disconnected and normal operation resumes when the communication to the Core Service is restored. The User can continue working on an open Batch or create a new Batch. Valid values are integers from 1 to 20. (default value: 1)

scanapp.client.volume_license_consume_warn_ratio(percentage)

Displays a warning to clients upon login, once the consumption of volume licenses has reached the predefined percentage. (0 disables the warning)

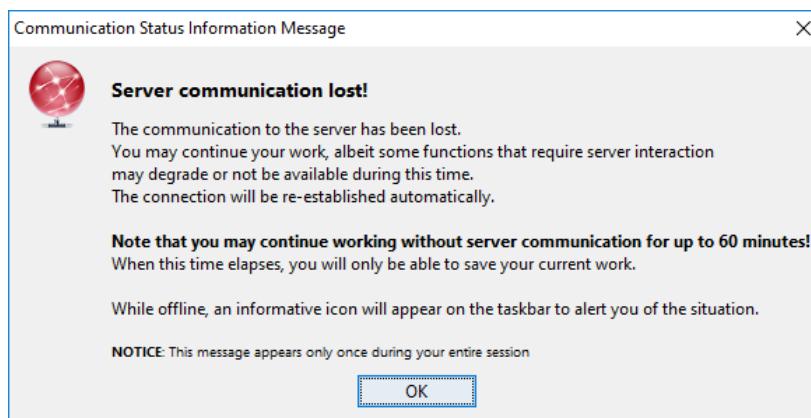


Figure 183. Server Communication Lost

12.2.2. Core Service parameters

allow_same_user_logins

Allow the same user account to start more than one concurrent sessions. When set to false, a new login attempt will disconnect any existing session. (default value: true)

audit.logging.enabled

Enable keeping an audit log of administrative actions, performed by any user with administrative rights. (default value: false)

stats.documents.enabled

Whether to keep statistics for documents in StatsBatchSession and StatsNodeSession tables. The default value is true. If this is set to false, no statistics on documents are retained.

batch.deletion.threshold.min

Threshold for temporarily keeping exported batches in the database. After this period, the exported batch will be deleted from the database, and the batch images will also be deleted from the repository. A value of zero deactivates the periodical check and deletion process. The default value is 1440 minutes (1 day). If it is set to zero then all finished batches will be deleted whenever the batch cleanup is executed. If it is set to a negative number, then finished batches are not deleted.

batch.cleanup.period.min

The default value is 60 minutes (1 hour). It determines how often the batch cleanup is executed. If it is set to zero, then the batch cleanup is deactivated. Note that batch cleanup also deletes batches that exceed the configured age limit (parameter batch.age.threshold).

delete_batches_permanently

When a batch deletion request is received at the server-side, this parameter determines whether the batches are permanently deleted from the database, or they will be marked as deleted but will remain available to batch / system administrators. (default value: true)

max.image.writers

Maximum Concurrent Image Writers. The default value is 0 (e.g. no limit). This parameter places a restriction on the number of concurrent image writes to disk. Its value should be such, so that the disk is a) not overrun and b) not underutilized. This parameter is directly related to the Queue Depth of the storage subsystem. A value of 2 is fine for a typical spinning single-disk storage system. Specifying a value of 0 (zero) removes the restriction, which is recommended for SSD disks or fast SAN/NAS systems. NOTE: This parameter is CRITICAL for the performance of the system in high-load usage scenarios and should be carefully adjusted.

scanapp.cluster.fail.heartbeats

Determines after how many heartbeat periods a server component is considered to be down. The default value is 6. The value of this parameter cannot be less than 2.

scanapp.cluster.heartbeat.period.sec

Determines how often (in seconds) server components perform heartbeat in order to report their status. The default value is 10. The period cannot be less than 10 sec.

batch.age.threshold

Batch Age Threshold in days. Batches that were created before the given number of days will be deleted, regardless of the step or mode that they are in. The default value is zero, which means that the deletion is deactivated. The check and deletion process will execute once every hour.

timestamp.difference.threshold.seconds

Certain requests sent by the clients include the client's timestamp. The sever may reject requests where the time difference between the client and the server is greater than the configured threshold. The threshold is given in seconds and the default value is 5 minutes, or 300 seconds. To disable the timestamp check, set the parameter to 0.

scanapp.cluster.health.check.period.sec

Determines how often (in seconds) server components perform health checks. The default value is 10 minutes (600 sec). The period cannot be less than 60 sec.

tesseract.base.dir

The directory under which the Tesseract archive files are found. The directory can be given as an absolute path, or as a path relative to the web application's path. If this property is empty, then the directory <installation directory>/add-ons/ocr-data will be used.

batch.folder.deletion.threshold.min

Orphan batch folder deletion threshold in minutes. If given a value greater than 0 the system will delete orphan batch folder that were last modified before the designated number of minutes.

12.3. Debug Client

A stand-alone version of the *Thick Client* application is available on the Info Input Solution application *Core Service*. The Info Input Solution administrator needs to login to the application server, to directly use the *Thick Client*. A shortcut link can be found in *Start menu* → *Info Input Solution* folder → *Info Input Solution Debug Client*.



Figure 184. Stand-alone Client login window

The difference between this stand-alone bundle and the *Thick Client* application , is that the admin-oriented stand-alone Client will not use the currently enabled java version on the application server. Instead, it is bundled with all required java files. Thus, it can be used to confirm the health and functionality of the Info Input Solution application server, before further investigating any problems caused by the user-specific browser security settings or java version related issues. The stand-alone Client user interface is the same with the *Thick Client*.

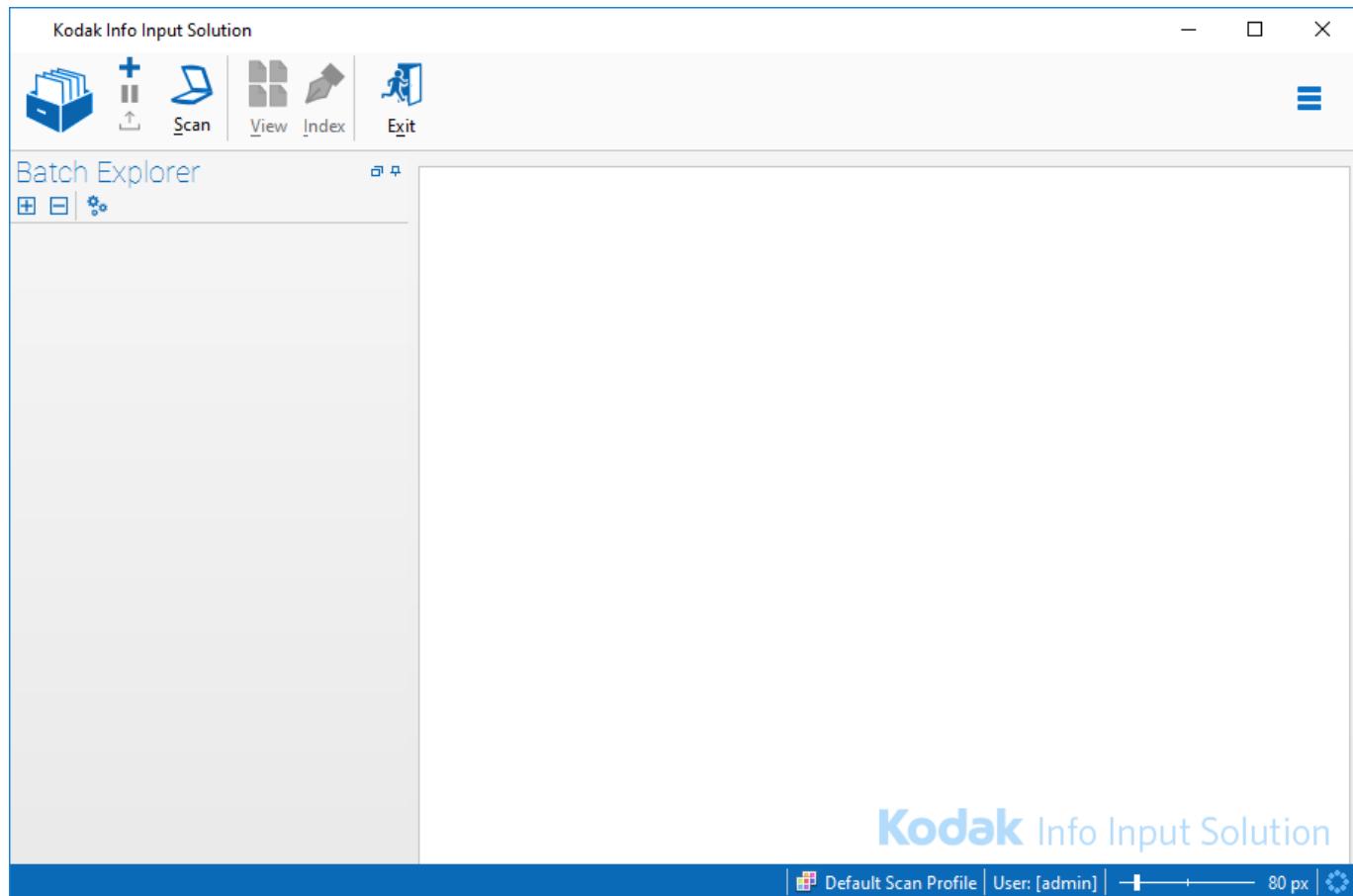


Figure 185. Stand-alone Client main window

12.4. Server Configuration properties

12.4.1. Overview

Starting from version 7.0, the `svc-core.properties`, `svc-html.properties`, `importd.properties` and `released.properties` files will be empty by default. This marks a shift in configuration management, allowing for customization according to one's specific needs. This documentation outlines the parameters available for configuration in these files.



The default values for all the parameters can be found in the following directory → `<your_installation_directory>\<released||importd||server||service-html>\WEB-INF\classes`. There, the corresponding template files can be spotted with the same name as the ones used in the config directory.

12.4.2. svc-core.properties

images.dir

Image Repository—the place where scanned documents are saved. This could be a path in the local file system or a network path. For clustered environments, this will normally reside on Network-Attached Storage.



If a network path is then the following syntax must be followed. \\\\\\\sharedstorage\\\\<directory>\\\\<subdirectory>... (8 slashes in the start, followed by 4 and so on)

async.startup (true/false)

Server asynchronous initialization and start-up. If given the value "true" the server will only perform a preliminary start-up at deployment or Web Container start-up time. Further initialization and start-up will be performed in the background. Useful for cases where the database server is not guaranteed to be online at the time the server starts up.



With asynchronous initialization enabled, the server IS NOT UP when deployment or container start-up complete. Server initialization is complete, and the server starts serving client requests when the message "Server started - SERVER IS UP" is written to the system log.

use.shortTermCache(true/false)

Decides whether short-term cache will be used.

imagenio

Set this property if the filesystem of the server has large latency during small writes. NFS/SMB systems usually have this issue. Available values are: 'native', 'wrapped', 'memoryheap', 'memory-offheap', 'memorycache'. The default value is 'native' for Windows, 'wrapped' for linux. All 'memory*' methods perform in-memory IO and will increase the memory requirements of the process.



Use this method only if the default (`native`) method performs significantly slower (e.g. less than 2-3 times slower) than the other methods. Use the script `tools/filesystem-performance-test.[sh|bat]` to run some tests on each server before considering modifying this property.

System Database Connection Pool settings

- `bonecp.partitionCount`
- `bonecp.maxConnectionsPerPartition`
- `bonecp.minConnectionsPerPartition`
- `bonecp.acquireIncrement`

- bonecp.closeConnectionWatch
- bonecp.closeConnectionWatchTimeout
- bonecp.idleMaxAge
- bonecp.idleConnectionTestPeriod
- bonecp.statementReleaseHelperThreads

dispatcher.resource.percentage & dispatcher.resource.count

These properties determine the number of workers (processes) that will be spawned to handle image operations. One of the two must be set. The '`dispatcher.resource.count`' has priority over the '`dispatcher.resource.percentage`'. If `[SVC-CORE]` is the only/main process on the machine, then a good approach is to set '`dispatcher.resource.percentage=75`' which means that 75% of CPU/core will be used. Alternatively the '`dispatcher.resource.count`' property could be set to the number of cores minus one.

dispatcher.max.worker.life.mins

Dispatcher worker processes can be terminated and restarted periodically. The following property determines how long the worker processes will be kept alive. The default value is 60 minutes and the minimum value is 10 minutes. Setting this property to 0 will cause the worker processes to stay alive indefinitely.

dispatcher.worker.startup.timeoutsecs

Timeout in seconds for starting and initializing a worker process. If the worker process does not respond within the given timeout, the dispatcher will terminate it. The default value is 30 seconds. The timeout cannot be less than 10 seconds.

worker.helper.params & dispatcher.debug

These are startup parameters for the workers (processes) spawned to handle image operations. You shouldn't change most of these. Usually there shall be no need for more than 1GB per process. Do not include double quotes in the params even if they include spaces.

server.script.threads

Number of threads dedicated to processing server script steps.



Any server script configurations should be bundled together and engulfed by the following: `Server Script setup: Begin` → Marks the start of the Server Script configuration. `Server Script setup: END` → Marks the end of the Server Script configurations.

extraction.threads

Number of threads dedicated to processing extraction steps.

registration.train.data.cache.expiry.mins

Expiration timeout in minutes for the registration train data cache. The cache is kept in the dispatcher worker processes to avoid repeatedly deserializing the data from the file on the disk.



Any extraction configurations should be bundled together and engulfed by the following: `Extraction setup: BEGIN` → Marks the start of the Extraction configuration. `Extraction setup: END` → Marks the end of the Extraction configurations.

classification.threads

Number of threads dedicated to classification.

classification.project.cache.expiry.mins

Expiration timeout in minutes for the classification projects cache. The cache is kept in the dispatcher worker processes to speed the classification of multiple images with the same classification project.



Any classification configurations should be bundled together and engulfed by the following: `Classification setup: BEGIN` → Marks the start of the Classification configuration. `Classification setup: END` → Marks the end of the Classification configurations.

ocr.dispatch.threads

Number of threads dedicated to dispatching OCR tasks (database bound)

ocr.io.threads

Number of threads for non-blocking IO.

ocr.cpu.threads

Number of threads for cpu-bound operations

Google DocumentAI specific

- `ocr.googledocai.queue.consumer.threads`: Number of threads for parallel processing of documents. The sum of this number for each cluster node must be less or equal than the maximum number of concurrent requests supported by the engine.
- `ocr.googledocai.retry.wait`: Number of SECONDS to wait between retries in case of a Resource Exhausted exception.
- `ocr.googledocai.max.retries`: Number of retries to perform while waiting for DocumentAI resources to be freed.

Amazon Textract specific

- `ocr.extract.queue.consumer.threads`: Number of threads for parallel processing of documents.

The sum of this number for each cluster node must be less or equal than the maximum number of concurrent requests supported by the engine.

- `ocr.texttract.max.retries`: Number of retries to perform while waiting for Textract resources to be freed.

Azure Form Recognizer specific

- `ocr.azureformrec.queue.consumer.threads`: Number of threads for parallel initial upload of documents. The sum of this number for each cluster node must be less or equal than the maximum number of concurrent requests supported by the engine.
- `ocr.azureformrec.retry.wait`: Number of SECONDS to wait between retries in case of a Resource Exhausted exception.
- `ocr.azureformrec.max.retries`: Number of retries to perform while waiting for Form Recognizer resources to be freed.

 Any Intelligent OCR configurations should be bundled together and engulfed by the following: `Intelligent OCR setup: BEGIN` → Marks the start of the Intelligent OCR configuration. `Intelligent OCR setup: END` → Marks the end of the Intelligent OCR configurations.

`imageenhancement.threads`

Number of threads dedicated to Image Enhancement.

 Any Image Enhancement configurations should be bundled together and engulfed by the following: `Image Enhancement setup: BEGIN` → Marks the start of the Image Enhancement configuration. `Image Enhancement setup: END` → Marks the end of the Image Enhancement configurations.

`extlib.dir`

Default directory for plugins. (default value is '`extlib`', below '`WEB-INF/`'). Each subfolder below this folder is considered a '`plugin`': all classes and jar files are loaded in a separate classloader.

`extlib.noparentclassloader.list`

A comma-list of sub-directories below the '`extlib.dir`' whose classloader should not delegate to the parent classloader (by default, the parent classloader is used if a class cannot be found in the plugin library's classpath) This is useful if a library from the parent classloader somehow conflicts with the libraries of the implementation: this is rare and only happens with libraries that might use dynamic configuration.

 Any External libraries configurations should be bundled together and engulfed by the following: `External libraries & code: BEGIN` → Marks the start of the External libraries configuration. `External libraries & code: END` → Marks the end of the

External libraries configurations.

Custom Workflow Actors

- actor.<name>.class
- actor.<name>.<property-name>



Using the above syntax you, one or more actors can be registered. Where <name> is a random name to reference the actor in this file. Any number of properties that are actor specific can be added.

12.4.3. svc-html.properties

images.dir

Absolute path to the disk directory that will be available to the HTML Service. Make sure that the provided path is accessible to the service itself. This could be a path in the local file system or a network path.



If a network path is then the following syntax must be followed. \\\\\\\sharedstorage\\\\<directory>\\\\<subdirectory>... (8 slashes in the start, followed by 4 and so on)

response.header.filter.config

Used in common-filter::ResponseHeaderManagerFilter; also defined in web.xml (legacy).

imagenio

Set this property if the filesystem of the server has large latency during small writes. NFS/SMB systems usually have this issue. Available values are: 'native', 'wrapped', 'memoryheap', 'memory-offheap', 'memorycache'. The default value is 'native' for Windows, 'wrapped' for linux. All 'memory*' methods perform in-memory IO and will increase the memory requirements of the process.



Use this method only if the default (native) method performs significantly slower (e.g. less than 2-3 times slower) than the other methods. Use the script `tools/filesystem-performance-test.[sh|bat]` to run some tests on each server before considering modifying this property.

- app.config.dir:
- images.dir
- allow.rpc.ops.in.unlicensed.mode
- session.timeout.minutes
- filesystem.fileprovider.root.path

- http.fileprovider.url.pattern
- send.stacktrace.to.clients-
- generic.error.message
- upload.worker.count
- upload.cleanup.thread.max.count
- upload.transit.scan.period.hours
- upload.queue.max.size
- upload.queue.low.size
- instance.name
- use.dispatcher
- server.it.url
- client.it.url
- same-site-filter.enabled

dispatcher related

- worker.helper.params
- dispatcher.resource.count
- dispatcher.resource.percentage
- dispatcher.worker.helper
- dispatcher.worker.logging.config
- dispatcher.log.server.port
- dispatcher.max.threads.per.worker
- dispatcher.max.worker.life.mins
- dispatcher.worker.startup.timeoutsecs
- dispatcher.debug
- tesseract.base.dir
- extlib.dir
- extlib.noparentclassloader.list

db properties

- db.type
- db.url
- db.username
- db.password
- h2.db.listen.port
- h2.kv.listen.port
- h2.db.recreate-on-error
- h2.kv.recreate-on-error

datasource properties

- datasource.partitionCount
- datasource.maxConnectionsPerPartition
- datasource.minConnectionsPerPartition
- datasource.acquireIncrement
- datasource.closeConnectionWatch
- datasource.closeConnectionWatchTimeout
- datasource.idleMaxAge
- datasource.statementReleaseHelperThreads

nuance related properties

- nuance.bin.path
- nuance.default.lang.codes
- nuance.default.lang.mode
- nuance.default.engine
- nuance.kernel.soft.timeout
- nuance.kernel.hard.timeout

clustering

- multicast-ip
- tcp-local-address
- tcp-discovery-port
- tcp-communication-port
- tcp-discovery-addresses

license and auditing

- license-administrator.url
- license-coordinator.url
- license-coordinator.enable
- license.server.url
- backup.license.server.url
- license.directory
- license.keys
- legacy.licensing
- audit.report.interval.sec
- audit.dir

12.4.4. importd.properties

Srv-Core Config

- importd.server.username
- importd.server.password
- importd.server.url
- extlib.dir: Default directory for plugins. (default value is 'extlib', below 'WEB-INF/'). Each sub-folder below this folder is considered a '[plugin](#)': all classes and jar files are loaded in a separate classloader.
- Extlib.noparentclassloader.list : A comma-list of sub-directories below the 'extlib.dir' whose classloader should not delegate to the parent classloader (by default, the parent classloader is used if a class cannot be found in the plugin library's classpath). This is useful if a library from the parent classloader somehow conflicts with the libraries of the implementation: this is rare and only happens with libraries that might use dynamic configuration.
- imagenio: Set this property if the filesystem of the server has large latency on small writes. NFS/SMB systems usually have this issue. Available values are: '[native](#)', '[wrapped](#)', '[memory-heap](#)', '[memoryoffheap](#)', '[memorycache](#)'. The default value is '[native](#)' for Windows, '[wrapped](#)' for linux. All '[memory*](#)' methods perform in-memory IO and will increase the memory requirements of the process.

imagenio

Set this property if the filesystem of the server has large latency during small writes. NFS/SMB systems usually have this issue. Available values are: '[native](#)', '[wrapped](#)', '[memoryheap](#)', '[memoryoffheap](#)', '[memorycache](#)'. The default value is '[native](#)' for Windows, '[wrapped](#)' for linux. All '[memory*](#)' methods perform in-memory IO and will increase the memory requirements of the process.



Use this method only if the default ([native](#)) method performs significantly slower (e.g. less than 2-3 times slower) than the other methods. Use the script [tools/filesystem-performance-test.\[sh|bat\]](#) to run some tests on each server before considering modifying this property.

Embedded Tomcat

- importd.embedded.http.serverroot
- importd.embedded.http.port
- importd.embedded.http.useSSL(true/false)= Required to enable SSL.
- importd.embedded.http.connector.maxThreads= Max number of threads to be used.
- importd.embedded.http.connector.proxyName
- importd.embedded.http.connector.proxyPort

SSL Key section



Add the following parameters to use ssl certificate and key. If added, the keystore section should be removed/commented out.

- importd.embedded.http.ssl_connector.clientAuth
- importd.embedded.http.ssl_connector.sslProtocol
- importd.embedded.http.ssl_connector.keystoreType
- importd.embedded.http.ssl_connector.keyAlias
- importd.embedded.http.ssl_connector.keystorePass
- importd.embedded.http.ssl_connector.keyPass
- importd.embedded.http.ssl_connector.keystoreFile

12.4.5. released.properties

released.page.image.directory

Image Storage Location. This could be a path in the local file system or a network path.



If a network path is then the following syntax must be followed. \\\\\\\sharedstorage\\\\\\<directory>\\\\\\<subdirectory>... (8 slashes in the start, followed by 4 and so on)

Batch polling settings

- released.batch.release.max.attempts
- released.batch.polling.interval
- released.batch.max.count

released.queue.low.count

This parameter determines the level which the queue must reach to execute the polling query to the DB. It can be a number from 0 up to '[released.batch.max.count-1](#)'. If the value is released.batch.max.count-1, then the polling query will be executed every time an item is removed from the queue. You may reduce the value of this parameter, to reduce the frequency with which the polling query is executed. Setting it to 0, means that the polling query will be executed only when the queue becomes empty. If no value (or invalid value) is given, then released.batch.max.count / 2 is used.

released.job.whitelist

You may define a comma separated list of job names, which will be the only jobs that this export server will process. If the whitelist is empty, then there will be no restriction to the jobs that this export server can process.

extlib.dir

Default directory for plugins(default value is '[extlib](#)'). Each subfolder below this folder is consid-

ered a '**plugin**': all classes and jar files are loaded in a separate classloader.

extlib.noparentclassloader.list

A comma-list of sub-directories below the '**extlib.dir**' whose classloader should not delegate to the parent classloader (by default, the parent classloader is used if a class cannot be found in the plugin library's classpath). This is useful if a library from the parent classloader somehow conflicts with the libraries of the implementation: this is rare and only happens with libraries that might use dynamic configuration.

Connection pool settings

- bonecp.partitionCount
- bonecp.maxConnectionsPerPartition
- bonecp.minConnectionsPerPartition
- bonecp.acquireIncrement

released.tesseract.base.dir

Tesseract OCR files directory (absolute path).

released.rubberstamps.archive.path

Define a custom archive file that contains rubberstamp images. By default, you don't need to set anything here, unless you are using pre-defined rubberstamps images in the clients.

Dispatcher properties

- dispatcher.resource.count: Defines the number of worker processes as an absolute number.
- dispatcher.resource.percentage: Defines the number of worker processes as a percentage of the available CPU cores. The default number of worker processes is 50% of the available CPU cores.
- dispatcher.max.worker.life.mins: Dispatcher worker processes can be terminated and restarted periodically. The following property determines how long the worker processes will be kept alive. The default value is 60 minutes and the minimum value is 10 minutes. Setting this property to 0 will cause the worker processes to stay alive indefinitely.
- dispatcher.worker.startup.timeout.secs: Timeout in seconds for starting and initializing a worker process. If the worker process does not respond within the given timeout, the dispatcher will terminate it. The default value is 30 seconds. The timeout cannot be less than 10 seconds.
- dispatcher.worker.jvm.options: Passes JVM options to the worker processes.
- dispatcher.worker.logging.config: Customizes the log4j for the worker processes.

Nuance properties

- nuance.bin.path: Nuance binaries path. Use this property to define the path where Nuance binaries are located. Normally you should not need to define this property. By default, in Windows platforms, the Nuance binaries are located under the add-ons directory, which is passed to the export service using the system property java.library.path. In Linux platforms, the Nuance binaries are located under /usr/local/lib.

- `nuance.searchable.pdf.soft.timeout.sec`: Soft timeout used when creating searchable PDFs with Nuance. It is given in seconds and the default value is 10 seconds. If Nuance takes longer than the given timeout to process a single page, then the operation will fail. If the export server runs on a slow or loaded machine, you may need to increase the timeout.

13. Intelligent REST Services

The Rest API service Administration menu can be accessed from the Tools & Options menu in the Thick Client by selecting the *Rest API Service Administration*.

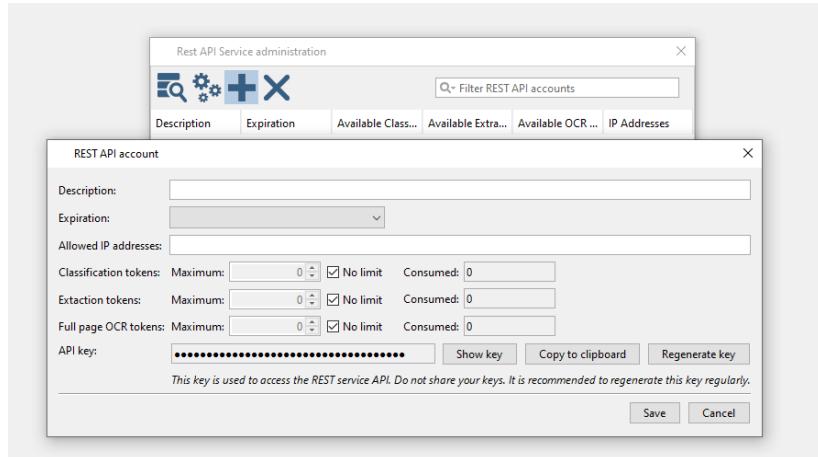


Figure 186. Rest API Service Administration Window

In this window, an API key can be created, to be used from a third party application, and its properties can be modified. The API key can be used so that the Info Input Solution components can be integrated and used from the third party application.

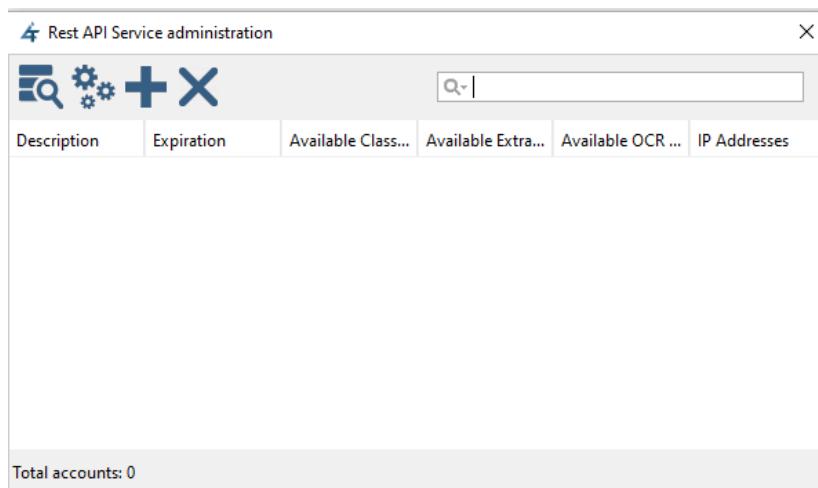


Figure 187. Rest API service Administration window

The key can be configured with multiple properties related to security such as expiration, allowed IP addresses and maximum consumption of License Tokens but the important value of the API key string itself which is what is required to create a session to start using the Intelligent REST Services.

For more information about the Intelligent Rest Services please refer to the corresponding section in

the Developer's guide.

Appendix A: Regex Search

A.1. Introduction

Regular expression (regex) searches are available for use in certain places in the application. For example, text matching in order to determine the document class type during classification.

The regular expression engine used is *TRE*. It features the ability to use "fuzzy search"; i.e. the ability to match text that is not exactly as expected. This comes in handy when the text input comes via OCR of images of potentially inferior quality, where some letters are miss-read.

Fuzzy search is automatically enabled whenever you use a regular expression and a "minimum confidence" that is below 100%.

A.2. Regular Expression Syntax

The regular expression syntax is POSIX "Extended RE" (ERE).

A.3. Limitations

The regular expression syntax is not Perl-compatible and has the following limitations:

- When fuzzy matching is enabled, the non-greedy modifier `?` is not supported
- Look-ahead and look-behind assertions are not supported

A.4. Technical Details

Approximate (i.e. "fuzzy") matching uses the Levenshtein distance (see the [Levenshtein distance article on Wikipedia](#)) to calculate a score for each match. The score is calculated using the following values:

- The cost for an insertion is "1"
- The cost for a deletion is "1"
- The cost for a replacement is "1"

The maximum cost for a match is 20, i.e. matches with a higher cost are discarded.

The confidence for a match is calculated by the following formula:

$$1 - (\text{<cost>} / \text{<original text length>})$$

Warning: Be careful with regular expressions that match shorter lengths of text. To avoid matching text that differs much more than intended, you must use a higher confidence setting. Prefer regular expressions that match a longer piece of text to avoid problems.